(45) Date of publication and mention of the grant of the patent:
24.05.2006 Bulletin 2006/21

(21) Application number: 00303745.4

(22) Date of filing: 04.05.2000

(54) Methods and apparatus for data processing, methods and apparatus for data reproducing and recording media
Datenverarbeitungsverfahren und -gerät, Datenwiedergabeverfahren und -gerät,
Datenaufzeichnungsmedien
Appareil et méthodes de traitement de données, méthodes et appareil de reproduction de données,
moyn d'enregistrement

(84) Designated Contracting States:
DE FR GB NL

(30) Priority: 06.05.1999 JP 12616499

(43) Date of publication of application:
08.11.2000 Bulletin 2000/45

(73) Proprietor: SONY CORPORATION
Tokyo 141 (JP)

(72) Inventors:
• Kato, Motoki awa,
  Shinagawa-ku,
  Tokyo 141 (JP)

• Hamada, Toshiya awa,
  Shinagawa-ku,
  Tokyo 141 (JP)

• Nakamura, Masanobu awa,
  Shinagawa-ku,
  Tokyo 141 (JP)

(74) Representative: Pilch, Adam John Michael et al
D Young & Co
120 Holborn
London EC1N 2DY (GB)

(56) References cited:
EP-A- 0 749 244
US-A- 5 754 651

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).
Description

[0001] The present invention generally relates to data processing methods/apparatus, data reproducing methods/apparatus, and recording media. More specifically, the present invention is directed to such data processing/recording methods/apparatus which may be capable of quickly random-accessing multiplexed program data by extracting feature point information from a coded transport stream.

[0002] In multi-channel digital television broadcasting systems known as the European DVB (Digital Video Broadcast) system and the Japanese digital BS (Broadcast Satellite) broadcast system, the MPEG (Moving Picture Experts Group) -2 transport stream is used. A transport stream corresponds to such a stream in which a transport packet is continued. A transport packet corresponds to such a packet that, for instance, either an MPEG-2 video stream or an MPEG-1 audio stream is packeted. Either one program or a plurality of AV (Audio Visual) programs are multiplexed on a single transport stream which is transmitted by way of broadcasting electromagnetic waves. In general, AV programs of each channel are independently provided from each other.

[0003] As a consequence, when a transport stream transmitted in a broadcasting channel is directly received and then recorded by a home-use receiver, AV programs of all channels contained in this transmitted transport stream may be recorded at the same time. Also, when transport streams of AV programs of several channels selected from this broadcasted transport stream by a user are separated and then recorded, AV programs of an arbitrarily selected channel may be recorded at the same time.

[0004] Fig. 1 indicates an example of one conventional transport stream recording method. Fig. 1A shows such a transport stream on which a plurality of AV programs is multiplexed. In this drawing, an abscissa shows time. This time is segmented every time unit TUi (i = 0, 1, 2, ---) having an interval “Δt”. Either one AV program or a plurality of AV programs are selected from an input transport stream. The selected transport packets are indicated by using hatched lines. Generally speaking, as shown in Fig. 1B, these selected transport packets will appear at irregular timing, and thus, a total number of transport packets every time unit TUi is changed.

[0005] As indicated in Fig. 2, the transport packets which are selected every such a time unit TUi having the interval of “Δt” are recorded on a recording medium, while the intervals are shortened. At this time, these transport packets are recorded on this recording medium, while adding thereto time stamps indicative of time instants on the respective transport streams. This time stamp is similar to such a TSP_extra_header having a length of 4 bytes. That is, this 4-byte TSP_extra_header is added to the transport packet which is defined in, for instance, the DV (Digital Video) format (Specification Of Consumer-Use Digital VCRs using 6.3 mm magnetic tape in HD digital VCR conference), and PART7-DVB Specifications of Consumer-Use Digital VCR).

[0006] In Fig. 2, an abscissa indicates an address representative of a byte position of a recorded transport stream. When the transport stream having the variable bit rate shown in Fig. 1B is entered, a recording apparatus inserts dummy data into this input transport stream, and then records the resulting data at a fixed recording rate, as indicated in Fig. 2.

As a consequence, a data amount with respect to a time elapse of the recorded transport stream may have a direct proportional relationship. In other words, assuming now that an amount of recorded data per a time unit is selected to be “x”, a byte position of head data of an n-th (n= 0, 1, 2, ---) time unit is equal to such a position obtained by multiplying x by n.

[0007] In this recording method, since the dummy data is inserted into the transport streams so as to achieve a constant recording rate, the recording efficiency of the transport stream is not so high. However, if such dummy data is not inserted, then the time elapses of the recorded transport streams is no longer directly proportional to the data amount of the file. As a result, in the case that such data located at a desired position on the time axis of the transport stream is accessed, there is a problem that the data access characteristic would be deteriorated.

[0008] Also, generally speaking, in an MPEG-2 video data stream, I picture are coded in an interval of approximately 0.5 seconds, and other pictures are coded as either P pictures or B pictures. As a consequence, when a video signal is reproduced in a high speed from a recording medium on which the MPEG-2 video data streams are recorded, such I pictures must be searched. However, in the case that the I pictures are reproduced by way of the random access manner from the recording medium on which the transport streams such as the digital broadcasting system are recorded, it is practically difficult to search the starting bytes of the I pictures in higher efficiencies. In other words, the syntax of the video streams which are read from the random byte positions of the transport streams recorded on the recording medium is analyzed, and thus, the starting bytes of either the I pictures or the audio frames are searched. As a consequence, lengthy search time of the I pictures is required in the worst case. Also, the high-speed random access reproducing operation cannot be carried out in response to the user input.

[0009] An aim of at least an embodiment of the present invention is to provide such data processing/recording methods/apparatus capable of quickly random-accessing multiplexed program data by extracting feature point information from a coded transport stream.

[0010] United States Patent No US-A-5 754 651 discloses a data processing apparatus for processing data contained in an entered coded stream (in accordance with the MPEG2 standard) comprising transport packets. In accordance with the MPEG2 standard, the transport packets relate to respective ones of a plurality of programs and the stream includes
Program Specific Information (PSI) for use in identifying and assembling individual transport packets to recover the content of selected programs. In the processing technique disclosed in US-A-5 754 651, a desired program is selected from the entered coded stream and condensed PSI (CPSI) is formed that includes program map information associating packet identifiers with individual packetised data streams that constitute the selected program and exclude program map information relating to other programs. A data stream constituted from the identified program content packets and the CPSI is stored.

[0011] European Patent Application Publication No EP-A-0 749 244, on which the pre-characterising parts of claims 1 and 17 hereof are based, discloses another arrangement which processes an entered MPEG2 coded stream with a view to separating transport packets that relate to a desired program. Selected transport packets are passed to a digital interface in which their times of arrival are noted and a time stamps are added to the transport packets to produce source packets.

[0012] European Patent Application Publication No EP-A-0 903 738 discloses the storage, on a DVD, of both AV data in the form of video objects and first and second time tables for each such object. The first time table includes addresses of video object units in a corresponding video object, the addresses being arranged in order that correspond to reproduction points that differ by a predetermined time unit longer than a maximum reproduction period of a video object unit, and indicators for specifying the video object units which respectively correspond to the addresses. The second time table includes an entry for each video object unit in the corresponding video object, the entries being arranged in order and each including a reproduction period and a data size of a video object unit.

[0013] The present invention provides:

- a data processing apparatus according to claim 1 hereof;
- a data processing method according to claim 17 hereof;
- a recording medium, having a program recorded thereon, according to claim 33 hereof;
- a data reproducing apparatus according to claim 34 hereof;
- a data reproducing method according to claim 36 hereof;
- a recording medium, having a program recorded thereon, according to claim 38 hereof; and
- a recording medium, having source packets and a time unit map recorded thereon, according to claim 40 hereof.

[0014] In a data processing apparatus, a data processing method, and a program of a first recording medium, according to a preferred form of implementation of the present invention described herebelow, the entered coded stream is segmented every time unit; and both the time unit map indicative of the address of the data for every such time unit, and also an entry point map subordinate to the time unit map and representative of the position of the entry point of the coded stream, are generated.

[0015] In a data reproduction apparatus, a reproducing method, and a program of a second recording medium, according to the preferred form of implementation of the present invention, both the time unit map indicative of the address of the data every time unit of the coded stream, and also the entry point map subordinate to the time unit map for indicating the position of the entry point of the coded stream, are produced from the recording medium. Then, the coded stream is reproduced from an arbitrary position based upon the time unit map and entry point map.

[0016] In a recording medium according to the preferred implementation of the present invention, the source packets of the coded stream and the time unit map indicative of the address of the data every time unit of the coded stream are recorded in the form of a file.

[0017] The invention will now be described by way of example with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

- Fig. 1 is an explanatory diagram for explaining the packet of the conventional transport stream;
- Fig. 2 is an explanatory diagram for explaining the conventional transport stream to be recorded;
- Fig. 3 is an explanatory diagram for explaining a transport stream according to an embodiment of the present invention;
- Fig. 4 is an explanatory diagram for explaining the transport stream to be recorded, according to the embodiment of the present invention;
- Fig. 5 is a diagram for indicating syntax of a source packet;
- Fig. 6 is a diagram for showing syntax of a TP_extra_header;
- Fig. 7 is an illustration for indicating an example of a time unit map;
- Fig. 8 is an explanatory diagram for explaining an offset address every time unit;
- Fig. 9 is a diagram for representing an example of an entry point map;
- Fig. 10 is an explanatory diagram for explaining entry point data;
- Fig. 11 is an explanatory diagram for explaining a delation of data;
- Fig. 12 is a diagram for indicating an example of a time unit map when data is deleted;
BASIC IDEA OF THE PRESENT INVENTION

First of all, a basic idea of the present invention will now be described. That is, when a moving picture recording apparatus, according to the present invention, records a transport stream on a recording medium such as a disk and a tape, this moving picture recording apparatus segments time on a transport stream with respect to a predetermined time unit (unit time), and then calculates an address on a data stream every time unit. In this transport stream, either one or a plurality of programs are multiplexed. Then, a time unit map is formed. This time unit map indicates the addresses on a data stream every this time unit. Furthermore, an entry point map is formed. This entry point map indicates a position of an entry point (namely, random access point) with respect of each program of a transport stream to be recorded. The entry point map owns a structure followed to the time unit map. Now, this time unit map will now be explained as follows:

Fig. 3 represents a transport stream according to the present invention; Fig. 10 is an explanatory diagram for explaining a relationship between a time unit map and an entry point map of a transport stream file; Fig. 11 is a schematic block diagram for showing a structural example of a moving picture recording apparatus to which the present invention is applied; Fig. 12 is a flow chart for describing a recording operation of the moving picture recording apparatus shown in Fig. 27; and Fig. 13 is a diagram for indicating syntax of a TimeUnitMapHeader;

[0018] Referring now to the drawings, various preferred embodiments of the present invention will be described.

[0019] A description of the present invention is made of such a case that a coded stream is a multiplexed stream on which either one program or a plurality of programs are multiplexed. It should be noted that the present invention may be similarly applied to the coded stream which is an elementary stream such as an MPEG video stream.

Fig. 17 is an explanatory diagram for explaining an entry point of a transport stream file; Fig. 18 is a diagram for showing syntax of an EntryPointMapData; Fig. 19 is an explanatory diagram for explaining an entry point of a transport stream file; Fig. 20 is a diagram for indicating an example of EntryPointMapData; Fig. 21 is a diagram for indicating an example of EntryPointMapData; Fig. 22 is an explanatory diagram for explaining a transport stream according to the present invention; Fig. 23 is a diagram for showing syntax of a TimeUnitMapHeader; Fig. 24 is a diagram for showing syntax of TimeUnitMapData; Fig. 25 is an explanatory diagram for explaining a deletion of data; Fig. 26 is a diagram for representing an example of a time unit map when data is deleted; Fig. 27 is a schematic block diagram for showing a structural example of a moving picture recording apparatus to which the present invention is applied; Fig. 28 is a flow chart for describing a recording operation of the moving picture recording apparatus shown in Fig. 27; Fig. 29 is a flow chart for describing a recording operation of the moving picture recording apparatus shown in Fig. 27; Fig. 30 is a flow chart for describing a recording operation of the moving picture recording apparatus shown in Fig. 27; Fig. 31 is an explanatory diagram for explaining a relationship between a time unit map and an entry point map of a transport stream file; Fig. 32 is a schematic block diagram for indicating another structural example of a moving picture recording apparatus to which the present invention is applied; Fig. 33 is a schematic block diagram for showing a structural example of a moving picture reproducing apparatus to which the present invention is applied; Fig. 34 is a flow chart for describing a reproducing operation of the moving picture reproducing apparatus shown in Fig. 33; Fig. 35 is a flow chart for describing a reproducing operation of the moving picture reproducing apparatus shown in Fig. 33; Fig. 36 is a flow chart for describing a reproducing operation of the moving picture reproducing apparatus shown in Fig. 33; Fig. 37 is a flow chart for describing a reproducing operation of the moving picture reproducing apparatus shown in Fig. 33; and Fig. 38 is an explanatory diagram for explaining a recording medium according to the present invention.

[0020] First of all, a basic idea of the present invention will now be described. That is, when a moving picture recording apparatus, according to the present invention, records a transport stream on a recording medium such as a disk and a tape, this moving picture recording apparatus segments time on a transport stream with respect to a predetermined time unit (unit time), and then calculates an address on a data stream every time unit. In this transport stream, either one or a plurality of programs are multiplexed. Then, a time unit map is formed. This time unit map indicates the addresses on the data stream every this time unit. Furthermore, an entry point map is formed. This entry point map indicates a position of an entry point (namely, random access point) with respect of each program of a transport stream to be recorded. The entry point map owns a structure followed to the time unit map. Now, this time unit 15 map will now be explained as follows:

Fig. 3 represents a transport stream formed by that a plurality of AV programs are multiplexed. In this drawing, an abscissa shows time, and the time is segmented every time unit TUi (i = 0, 1, 2, ...) having a time interval Δt. A magnitude of Δt is selected to be, for instance, 0.5 seconds. Either one AV program or a plurality of AV programs are selected from an input transport stream so as to be recorded. The selected transport packets are indicated by applying hatched lines thereto. In general, as shown in Fig. 3B, the selected transport packets appear at irregular timing, and a total number of transport packets every time unit TUi having the time interval Δt is changed. It should be understood that in

[0021] A magnitude of Δt is selected to be, for instance, 0.5 seconds. Either one AV program or a plurality of AV programs are selected from an input transport stream so as to be recorded. The selected transport packets are indicated by applying hatched lines thereto. In general, as shown in Fig. 3B, the selected transport packets appear at irregular timing, and a total number of transport packets every time unit TUi having the time interval Δt is changed. It should be understood that in
an actual transport stream, considering now that a time length of a time unit is substantially equal to 0.5 seconds, an actual number of the transport packets contained in this time unit becomes very larger than that of the example shown in Fig. 3. However, in this embodiment mode, these transport packets are represented in a simple manner.

[0022] As indicated in Fig. 4, the selected transport packets are recorded on a recording medium (not shown), while intervals among these selected transport packets are shortened. At this recording operation, time stamps (Arrival Time Stamps) indicative of time instants on the transport stream are added to the respective transport packets. It is now assumed that an Arrival Time Stamp is similar to, for example, a TSP_extra_header having a length of 4 bytes, which is added to a transport packet defined by a DV format. In this specification, such a transport packet to which a header having a length of 4 bytes and containing this Arrival_Time_Stamp is added will be referred to as a "source packet."

Since a transport packet owns a length of 88 bytes, this source packet owns a length of 192 bytes.

[0023] As shown in Fig. 5, syntax of a source packet is constituted by both a TP_extra_header () and a transport_ packet ( ). As shown in Fig. 6, this TP_extra_header () is arranged, by both a copy_permission_indicator and an arrival_time_stamp.

[0024] In Fig. 4, an abscissa is an address indicative of a byte position of a recorded transport stream. Also, the abscissa shows a head address of such a transport packet which is firstly inputted every time unit on the abscissa. In this example, 4 pieces of transport packets are contained in a time unit TU0, 3 pieces of transport packets are contained in a time unit TU1, and 6 pieces of transport packets are contained in a time unit TU2. A transport packet which is inputted by bridging two time units is contained in a front-sided time unit. It is also assumed that head addresses of firstly-entered transport packets, of these time units TU0, TU1, TU2, or head addresses of source packets containing these transport packets are indicated as A(TU0), A(TU1), and A(TU2), respectively.

[0025] Fig. 7 indicates a time unit map, namely, an example of a table of head addresses of every time unit of a recorded transport stream. In this case, a time_unit_address indicates an address of head data of a time unit on a recorded stream. In the time unit map, data lengths delta_time_unit_address every time unit are formed in a table.

[0026] In this example, a data length of the time unit TU0 is expressed by a difference [A(TU1) - A(TU0)] between the head address A(TU1) of the time unit TU1 and the head address A(TU0) of the time unit TU0. Similarly, a data length of the time unit TU1 is expressed by a difference [A(TU2) - A(TU1)] between the head address A(TU2) of the time unit TU2 and the head address A(TU1) of the time unit TU1. Similarly, a data length of the time unit TU2 is expressed by a difference [end_address_A(TU2)] between the end address end_address of the time unit TU2 and the head address A(TU2). The time unit TU2.

[0027] Next, the above-explained entry point map is explained. A transport stream shown in Fig. 8 is such a transport stream similar to the transport stream shown in Fig. 4. In this case, it is now assumed that in a transport packet indicated by a hatched line, an entry point is commenced. Concretely speaking, in the entry point, it is so assumed that a sequence header and I picture data of MPEG video data are commenced. In such a case that an entry point is present in a preselected time unit, an offset address is calculated which is defined from a head address of data of this predetermined time unit up to an address of an entry point. In other words, in the example of Fig. 8, entry points (I picture) are present in the time unit TU0 and the time unit TU2. Thus, in the time unit TU0, an interval "a" is calculated as an offset address. This interval "a" is defined from a head address A(TU0) of this time unit TU0 up to a head address l_start_address of the I picture. At the same time, in the time unit TU2, an interval "b" is calculated as an offset address. This interval "b" is defined from a head address A(TU2) of this time unit TU2 up to a head address l_start_address of the I picture.

[0028] Fig. 9 indicates an entry point map, namely an example of a table of an offset address up to an entry point every time unit. A flag entry_point_flag is set to "1" when an entry point is present in the corresponding time unit "TU1." This flag entry_point_flag is set to "0" when an entry point is not present in the corresponding time unit. As to such a time unit whose entry_point_flag is "1", an offset address l_start_offset_from_time_unit_address is calculated by the following formula, which is defined from a head address time_unit_address of data in this time unit up to an address l_start_address of an entry point:

\[
\text{l_start_offset_from_time_unit_address} = \text{l_start_address} - \text{time_unit_address}
\]

[0029] Also, with respect to each of the entry points, a calculation is made of: an end address l_end_address of an I picture data of the entry point; an end address P1_end_address of one next P picture, or one next I picture of the I picture of the entry point; and an end address P2_end_address of two next P picture, or two next I picture of the I picture of the entry point in accordance with the below-mentioned formula:

\[
\text{l_end_offset_address} = \text{l_end_address} - \text{l_start_address} \\
\text{P1_end_offset_address} = \text{P1_end_address} - \text{l_start_address} \\
\text{P2_end_offset_address} = \text{P2_end_address} - \text{l_start_address}
\]
[0030] Fig. 10 shows a concrete example of these addresses. That is, Fig. 10 represents MPEG video data starting from a head of a preselected time unit. In this drawing, symbols I, P, B indicate an I picture, a P picture, and a B picture, respectively. Also, suffixed numerals show a display order of these pictures. In this time unit, there is an I picture of an entry point indicated by "I2." Also, a P picture subsequent to the I picture I2 corresponds to "P5", and a P picture two subsequent to this I picture corresponds to "P8." At this time, the addresses of I_start_offset_from_time_unit_address; I_end_offset_address; P1_end_offset_address; and P2_end_offset_address calculated by the above-explained formula establish such a relationship as indicated in this drawing.

[0031] In other word, the address I_end_offset_address is assumed as such a value obtained by subtracting the starting address I_start_address of the I picture I2 from the end address I_end_address of the I picture I2. Also, the address P1_end_offset_address is assumed as such a value obtained by subtracting the starting address I_start_address of the I picture I2 from the end address P1_end_address of the P picture P5. Further, the address P2_end_offset_address is assumed as such a value obtained by subtracting the starting address I_start_address of the I picture I2 from the end address P2_end_address of the P picture P8.

[0032] The address I_start_offset_from_time_unit_address is assumed as such a value obtained by subtracting the head address time_unit_address of the data of the time unit from the address I_start_address of the entry point.

[0033] It should be noted that when a plurality of programs are contained in the transport stream to be recorded, the information of the entry points is discriminated from each other every program to be formed. Considering such a case that the entry point data could not be prepared for all of the programs, the entry point map contains such information (parsed_program_flag) for indicating as to whether or not the entry point data is present every program.

[0034] In the case that the transport streams recorded on the recording medium are edited, the time unit map thereof is changed (updated). Next, this changing method will now be explained. Fig. 11A represents an example of such a case that both the 2 head packets of the transport stream shown in Fig. 4 and the 3 end packets thereof are deleted. Fig. 11B shows such a transport stream produced after the packets are partially deleted in this manner. Fig. 12 represents a time unit map of the transport stream shown in Fig. 11B. When the data contained in a half way of the time unit are deleted in this manner, since a time length (first_time_unit_size) of the first time unit Tu0 is changed, this is rewritten. In the case of Fig. 11B, the length of the unit Tu0 is changed into a difference value between a time stamp of a head packet Pb of the time unit Tu1 and a time stamp of a head packet Pa of the time unit Tu0 after deletion. Also, as indicated in Fig. 12, an address delta_time_unit_address of the time unit Tu0 is updated as a difference value (A(TU1)-C) between an address A(TU1) of the head packet Pb of the time unit Tu1 and an address C of the head packet Pa of the time unit Tu0 after deletion. When the time unit map is changed, the entry point map related to this time unit map is also changed.

[0035] Fig. 13 and Fig. 14 indicate an example of syntax of the above-explained time unit map. Fig. 13 and Fig. 14 represent a header portion [TimeUnitMapHeader( )] and a data portion [TimeUnitMapData( )] of the time unit map, respectively. When the time unit map is recorded as a file, both the header portion and the data portion may be recorded as a single file, or as separate files, respectively. Symbol "start_time" and symbol "end_time" of TimeUnitMapHeader( ) denote a starting time instant and an end time instant of this time unit map, respectively. For instance, these symbols indicate a recording start time instant and a recording end time instant when a certain transport stream is recorded. Symbol "first_time_unit_size" indicates a time length of a first time unit. Symbol "time_unit_size" shows a time length of a second time unit, or of a time unit after the second time unit. Symbol "number_of_time_unit_entries" represents a total number of time units contained in the transport stream. The addresses delta_time_unit_address (see Fig. 7) whose number is indicated by this symbol "number_of_time_unit_entries" are written in the data unit TimeUnitMapData( ).

[0036] Also, a first example of the syntax of the above-described entry point map is indicated in Fig. 15 to Fig. 17. Fig. 15 shows a header portion (EntryPointMapHeader( ) of an entry point map, and Fig. 16 indicates a data portion [EntryPointMapData( )] of an entry point map. Fig. 17 further indicates syntax of entry_point_data( ) shown in Fig. 16. When the entry point map is recorded as a file, both the header portion and the data portion may be recorded as a single file, or separate files, respectively.

[0037] Symbol "number_of_programs" of the header EntryPointMapHeader( ) shown in Fig. 15 indicates a total number of programs contained in the transport stream. There is such information indicating as to whether or not an entry map table is present with respect to each of programs to be recorded in lines of this syntax from a third line to a sixth line. The program number program_number of the fourth line is such an information specifying (discriminating) a program, and is equal to information which is written in a PMT (Program Map Table) of the corresponding program. The flag parsed_program_flag of the fifth line is such a flag indicating as to whether or not there is entry point data of this program.

[0038] The information about PMTs of the respective programs to be recorded are continued from an eighth line to a tenth line. Symbol "MPEG2_TS_program_map_section( )" indicates a PMT which is extracted from a transport stream to be recorded, and is defined based upon the MPEG-2 system rules. In this case, symbol "NUMBER_OF_ParsedPrograms" indicates a total number of programs whose parsed_program_flag is equal to "1". A sequence order of data appeared in a loop of the number NUMBER_OF_ParsedPrograms in the eight line corresponds to a sequence order of the numbers program_number whose parsed_program_flag is equal to "1" in a loop of the program number of programs
In EntryPointMapData( ) of Fig. 6, data of an entry point as to each of programs to be recorded is described. As a parameter of an entry point with respect to one time unit, there are both "entry_point_flag" and "entry_point_data ( )." As indicated in Fig. 17, the contents of entry_point_data( ) as to one time unit are "entry_point.time_stamp", "I.start_offset_from.time_unit_address", "I.end_offset_address", "P1_end_offset_address", and "P2_end_offset_address." In this case, entry_point.time_stamp is calculated based upon either a time instant at a stream of a transport packet of an entry point or a PTS (Presentation Time Stamp) of an I picture of an entry point. PTS corresponds to such information added to a header of a PES packet defined in the MPEG-2 system rules.

Fig. 18 shows a second example of the syntax of the above-explained entry point map. Both a structure of EntryPointMapHeader( ) and a structure of entry_point_data( ) are similar to those of the above-explained first example as indicated in either Fig. 15 or Fig. 17. As apparent from a comparison between Fig. 18 and Fig. 16, data arrangement orders of entry points as to each of programs of the second example are different from those of the first example shown in Fig. 16.

Next, data arrangement orders of entry maps are shown in the first example and the second example under the below-mentioned conditions. In this case, as shown in Fig. 19, it is now assumed that 3 pieces of programs (program #1, program #2, program #3) contained in a transport stream are multiplexed, and an entry point of each of programs is present every time unit TUi (i=0,1,2,3). In this case, the respective parameters are given as follows:

\[
\begin{align*}
\text{number_of_time_unit_entries} & = 4 \\
\text{number_of_programs} & = 3 \\
\text{program_number} & = 1: \text{parsed_program_flag} = 1 \\
\text{program_number} & = 2: \text{parsed_program_flag} = 1 \\
\text{program_number} & = 3: \text{parsed_program_flag} = 1 \\
\text{NUMBER_OF_ParsedPrograms} & = 3
\end{align*}
\]

Fig. 20 shows an entry point map in the case of the first example (example of Fig. 16). In this case, the lists of the entry point data every program are formed in a separate form. In other words, as shown in Fig. 20A, EntryPointMapData of the program program #1 is arranged as follows. Since there is no entry_point_data #1-1 through entry_point_data #4 present as entry_point_data in each of the time units TU0 to TU3, each of the flags entry_point_flag is set to "1," and furthermore, entry_point_data #1-1 through entry_point_data #1-4 are described in the time units TU0 to TU3.

It should be noted that symbol "entry_point_data A-B" indicates entry_point_data( ) as to a B-th entry point of program_number = A.

As indicated in Fig. 20B, EntryPointMapData of the program #2 is arranged as follows. Since there is no entry_point_data in the time units TU1 and TU3, the flag entry_point_flag thereof is set to "0." To the contrary, since both entry_point_data #2-1 and entry_point_data #2-2 are present in each of the time units TU0 and TU2, the flag entry_point_flag thereof is set to "1," and furthermore, entry_point_data corresponding thereto, namely both entry_point_data #2-1 and entry_point_data #2-2 are described in the time units TU0 and TU2.

In addition, as to EntryPointMapData of program #3, since there is no entry_point_data in the time units TU0 and TU2, the flag entry_point_flag thereof is set to "0." To the contrary, since both entry_point_data #3-1 and entry_point_data #3-2 are present in each of the time units TU1 and TU3, the flag entry_point_flag thereof is set to "1," and furthermore, entry_point_data corresponding thereto, namely both entry_point_data #3-1 and entry_point_data #3-2 are described in the time units TU1 and TU3.

These entry_point_flag and entry_point_data are described in EntryPointMapData.

Fig. 21 indicates an entry point data in the case of the second example (example of Fig. 18).

In this case, entry point data of each of programs are arranged in a time order form every time unit, and a list of these entry point data constitutes a single form. In other words, in the time unit TU0, three sets of programs program #1 to program #3 are described, and entry_point_data corresponding to entry_point_flag is described in each of these three programs. In this example, since entry_point_data is not present in program #3, the flag entry_point_flag thereof is set to "0." To the contrary, since both entry_point_data #1-1 and entry_point_data #2-1 are present as to program #1 and program #2, the flag entry_point_flag thereof is set to "1."

Also, in other time units TU1 to TU3, both entry_point_flag and entry_point_data are described in each of programs #1 through program #3.

Referring now to Fig. 22, another example of a time unit map will be explained. Fig. 22A and Fig. 22B show arrangements similar to those of Fig. 3A and Fig. 3B as explained above. Also, similar to Fig. 4, Fig. 22C shows source packet data recorded on the recording medium. A relationship between a transport packet and a source packet is represented in Fig. 4. In Fig. 22C, an abscissa indicates the number of recorded source packet. The source packet number corresponds to an order at which the source packets are recorded. This source packet number is commenced from zero, and is sequentially incremented by 1. The head addresses A(TU0), A(TU1), and A(TU2) of the data every
time unit become 0, 4, 7, respectively, while being counted in the unit of source packet number.

[0051] Fig. 22D indicates a time unit map, namely, an example of a table for a head address of data every time unit of a recorded transport stream. In this example, symbol "RSPN (Relative Source Packet Number)_time_unit_start" shows an address of head data of a time unit on the recorded stream. In the time unit map, the addresses RSPN_time_unit_start are formed in a table in the order of time units from the head of the recorded transport stream. In the case of Fig. 22D, the address values are arranged in the order of A(TU0), A(TU1), A(TU2).

[0052] Next, an example of syntax of the above-explained time unit map are indicated in Fig. 23 and Fig. 24. Fig. 23 and Fig. 24 indicate a header portion [TimeUnitMapHeader()] of the time unit map, and a data portion [TimeUnitMapData()] thereof. When the time unit map is recorded as a file, both the header portion and data portion may be recorded as a single file, or may be alternatively recorded as separate files.

[0053] Symbol "offset_SPN" of the header portion TimeUnitMapHeader() shown in Fig. 23 indicates the number of the first source packet of the recorded transport stream. In this embodiment mode, symbol "RSPN_time_unit_start" is expressed by a relative source packet number with respect to the source packet number offset_SPN. An initial value of the source packet number offset_SPN when the time unit map is firstly formed is equal to zero. The next symbol "offset_time" represents a starting time instant of this time unit map, and indicates a starting time of a first time unit.

[0054] Symbol "time_unit_size" shows a time length (size) of the time unit of this time unit map.

[0055] Symbol "number_of_time_unit_entries" denotes a total number of time units within the recorded transport stream.

[0056] The addresses RSN_time_unit_start (Fig. 22D) whose number is indicated by the time unit number number_of_time_unit_entries are written in the data portion TimeUnitMapData() of Fig. 24.

[0057] Next, a description will now be made of a method for changing (updating) the above-explained time unit map in the case that the recorded transport stream is edited. Fig. 25A indicates such an example that two head packets of the transport stream shown in Fig. 22C are deleted, namely the packets defined from the head packet up to a source packet at an address indicated by "C" are deleted. Fig. 25B indicates a transport stream formed after the packets are partially deleted in this manner.

[0058] Fig. 26 shows time unit maps of transport streams in the case of Fig. 25A and Fig. 25B. As seen from this drawing, when the data located at the head portion of the time unit are deleted, data of RSPN_time_unit_start for referring to the time unit containing the deleted data portion is deleted. In the case of Fig. 25B, the data of RSPN_time_unit_start of the time unit TU0 is deleted. In connection thereto, the number "number_of_time_unit_entries" is decremented by a total number of data of the deleted RSPN_time_unit_start.

[0059] Also, the above-described offset_SPN and offset_time are changed. In the case of Fig. 26A, offset_SPN is equal to zero and also the starting time instant (namely 0) of the time unit TU0 is set to offset_time. In the case of Fig. 26B, offset_SPN is changed into the number C (= 2) of the original source packet for such a source packet which newly becomes a head packet. Also, such a time unit that offset_time newly becomes a head is changed, namely, in this example, this time unit is changed into the original starting time "ΔT" of the time unit TU1. It should also be noted that time_unit_size is not changed before/after the editing operation is carried out.

[0060] When the time unit map is changed, the entry point map related to this time unit map is also changed.

ARRANGEMENT OF MOVING PICTURE RECORDING APPARATUS

[0061] Next, Fig. 27 schematically shows a structural example of a moving picture recording apparatus 1 according to a preferred embodiment mode of the present invention. This moving picture recording apparatus 1 produces the above-explained table from an inputted transport stream, and records this produced table on the recording medium in combination with this input transport stream.

[0062] Either one AV program or a plurality of AV programs are multiplexed on a transport stream entered from a terminal 10 of this moving picture recording apparatus 1. On the other hand, a channel (service name) of such an AV program selected by a user interface is entered. In this case, a total number of selected channels may be selected to be 1, or more.

[0063] A PID filter 11 filters out a transport packet of a PID (Packet ID) designated by a stream analyzing unit 12 from the entered transport stream. Symbol "PID" indicates a code having a length of 13 bits, which is located at a fixed position of a header of the transport packet. This PID code represents a type of data stored in a payload of this transport packet. First, the PID filter 11 filters out a transport packet of a PAT (Program Association Table) in which PID=0×0000. In this PAT, a PID of a transport packet of a PMT (Program Map Table) of each of the programs which are multiplexed on the transport stream is described. The transport packet of the PAT filtered out from the PID filter 11 is entered into the stream analyzing unit 12.

[0064] A counter 24 counts a total number of packets defined from a head packet of a transport stream to be recorded up to a present packet. Then, the counter 24 outputs the present packet number to both a time unit map forming unit 23 and an entry point map forming unit 16.
The stream analyzing unit 12 extracts a PCR (Program Clock Reference) from such a transport packet which transfers this PCR, and then supplies the extracted PCR to a PLL unit 13. In such a case that there are plural PIDs of the transport packets which transfer PCRs, the stream analyzing unit 12 extracts PCR from any one of these PIDs of transport packets. The PLL unit 13 produces a clock having a frequency of 27 MHz in synchronism with the entered PCR, and then outputs this produced clock to a time stamp generating unit 14.

The time stamp generating unit 14 counts the entered clock, and generates a time stamp (Arrival_Time_Stamp) corresponding to this count value. Assuming now that the time stamp of the firstly recorded transport packet is set to zero, this counted time stamp may indicate time lapse after this transport stream is recorded. This time stamp is supplied to the stream analyzing unit 12, a time stamp adding unit 15, and a time unit map forming unit 23.

The time stamp adding unit 15 adds a header to the transport packet entered from the PID filter 11, and outputs the source packet (see Fig. 4) to a file system unit 17. This header contains a time stamp indicative of an arrival time instant of this entered transport packet.

The stream analyzing unit 12 forms entry point data of a stream to be recorded, and then inputs the formed entry point data to the entry point map forming unit 16. The content of this entry point data is indicated in Fig. 17. In such a case that the time stamp of the entry point is equal to a PTS of the entry point, since the PTS is derived from the input entry point data to the entry point map forming unit 16. The formed time unit map is outputted to both an entry point map forming unit 16 and the file system unit 17.

The stream analyzing unit 12 supplies the below-mentioned program information with respect to each of the programs to the entry point map forming unit 16:

1. program_number of a program;
2. a PID of a transport packet of a PMT of a program;
3. both a PID and stream_type of a transport packet of video data which constitutes a program;
4. both a PID and stream_type of a transport packet of audio data which constitutes a program; and
5. a PID of a PCR of a program.

In this program information, symbol "stream_type" implies a content written in a PMT. In the case of video data, this symbol "stream_type" indicates a stream type of video data such as MPEG-2 type video data and MPEG-1 type video data. In the case of audio data, this symbol "stream_type" denotes a stream type of audio data such as MPEG-1/AC-3 type audio data.

Also, the stream analyzing unit 12 forms entry point data of a stream to be recorded, and then inputs the formed entry point data to the entry point map forming unit 16. The content of this entry point data is indicated in Fig. 17. In such a case that the time stamp of the entry point is equal to a PTS of the entry point, since the PTS is derived from the input stream by the stream analyzing unit 12, the time stamp formed by the time stamp generating unit 14 need not be entered into the stream analyzing unit 12.

The entry point map forming unit 16 processes the entry point data to form an entry point data table every program, so that the above-described entry point map is formed and then is outputted to the file system unit 17.

RECORDING OPERATION OF MOVING PICTURE RECORDING APPARATUS

A description will now be made of recording operations of the above-explained moving picture recording apparatus. When a transport stream is entered from the terminal 10, the PID filter 11 employed in the moving picture recording apparatus filters out such a transport packet containing a PID (namely, PID=0×0000), and then outputs this extracted transport packet to the stream analyzing unit 12. At this time, the stream analyzing unit 12 executes a process operation described in a flow chart of Fig. 28.

When the stream analyzing unit 12 receives the transport packet of PID=0×0000 from the PID filter 11 at a step S11, the stream analyzing unit 12 acquires from a PAT thereof, a PID of a transport packet of a PMT of each of programs which is instructed via a terminal 22.

At a step S12, the stream analyzing unit 12 sets the PID of the PMT of each of the programs to the PID filter 11. When this PID filter 11 derives such a transport packet having PIDs of these PMTs, this transport packet is outputted to the stream analyzing unit 12.

At a step S13, the stream analyzing unit 12 receives the transport packet of a PMT from the PID filter 11. In this PMT, a PID of such a packet is written. That is, this packet transfers both the PID and a PCR (Program Clock Reference) of a transport packet which owns both a video stream and an audio stream, as a payload. These video and audio streams constitute this program. The stream analyzing unit 12 acquires the PID of such a packet which transfers both the PID and the PCR of a transport packet which owns the video stream and the audio stream as the payload at this step. These video stream and audio stream constitute each of the programs selected by a user interface.

At a step S14, the stream analyzing unit 12 sets to the PID filter 11, the PID of such a packet which transfers both the PID and the PCR of the transport packet which owns the video stream and the audio stream as the payload. These video stream and audio stream constitute each of the programs selected by the user interface.
It should be understood that when PID(s) (packet IDs) of a packet of service information used to transfer EPG (Electrical Program Guide) and the like are recognized in advance, these PID(s) are also set to the PID filter 11, and the packet of these PID(s) is also filtered out from this PID filter 11.

Then, the transport packet which is extracted by the PID filter 11 in this manner is supplied to the counter 24, the stream analyzing unit 12, and the time stamp adding unit 15. The counter 24 counts a total number of packets defined from a head packet of a transport stream to be recorded and a present packet, and then senses the present packet number. This sensed present packet number is supplied by this counter 24 to both the time unit map forming unit 23 and the entry point map forming unit 16.

Also, the stream analyzing unit 12 extracts a PCR (Program Clock Reference) from the entered transport packet, and then supplies the extracted PCR to the PLL unit 13. The PLL unit 13 generates the clock having the frequency of 27 MHz in synchronism with the entered PCR, and supplies this clock to the time stamp generating unit 14.

The time stamp generating unit 14 counts the entered clock to produce a time stamp corresponding to this count value. The time stamp adding unit 15 adds the time stamp to the transport packet entered from the PID filter 11, and then supplies the source packet to the file system unit 17. This time stamp indicates an arrival time instant of this transport packet, and is generated from the time stamp generating unit 14.

The time unit map forming unit 23 forms the time unit map based upon both the packet number entered from the counter 24 and the time stamp entered from the time stamp generating unit 14, and then supplies the formed time unit map to the entry point map forming unit 16 and the file system unit 17. In this time unit map, as indicated in Fig. 7, the addresses time_unit_address every time unit are made in correspondence with the addresses delta_time_unit_address every time unit. Also, the time unit map forming unit 23 forms another time unit map as shown in Fig. 22D, and then supplies this time unit map to the entry point map forming unit 16 and the file system unit 17. In this time unit map, the head data addresses RSPN_time_unit_start are made in correspondence with the time units.

Further, the stream analyzing unit 12 supplies the above-explained program information every program to the entry point map forming unit 16.

As a result, the stream analyzing unit 12 executes a process operation for analyzing an entry point as explained in flow charts shown in Fig. 29 and Fig. 30.

At a step S31 of this analyzing process flow chart shown in Fig. 29, the stream analyzing unit 12 sets both a PID of video data of a program to be recorded, and stream-type thereof to the PID filter 11. As a result, the designated packet of the video data is supplied from the PID filter 11 to the stream analyzing unit 12.

At a step S32, the stream analyzing unit 12 initializes a pointer "vpp" of a video packet, namely vpp = 0. This pointer vpp indicates an order (sequence) of a video packet of the above-described PID under process.

At a step S33, the stream analyzing unit 12 increments the pointer vpp of the video packet, for instance, increments this pointer by 1.

At a step S34, the stream analyzing unit 12 checks as to whether or not a code sequence-header-code of MPEG video data (namely, code having a bit length of 32 bits and indicated by "0X000001B3") is contained in a stream of a payload. When this code sequence_header_code is not contained in the stream, this video stream analyzing process operation is returned to the previous step S33.

To the contrary, when the stream analyzing unit 12 judges at the step S34 that the code sequence_header_code is contained in the payload, this analyzing process operation is advanced to a step S35. At this step S35, the stream analyzing unit 12 sets an address of a packet (namely, packet of first I picture) containing this code sequence_header_code to an address "I_start_address" (see Fig. 10).

At a step S36, the stream analyzing unit 12 increments the pointer vpp of the video packet.

At the next step S37, the stream analyzing unit 12 checks as to whether or not the data of the above-described I picture is ended. When the data of this I picture is not yet accomplished, the analyzing process operation is returned to the step S36. When the data of the I picture is ended, the analyzing process operation is advanced to a further step S38.

At this step S38, the stream analyzing unit 12 sets an address of a packet when the I picture is ended to "I_end_address" (see Fig. 10). With execution of the above-described analyzing process operation, the address of the first I picture is determined.

At a step S39 (while video pointer vpp is not incremented), the stream analyzing unit 12 checks as to whether or not a next video packet contains a sequence header code. When the stream analyzing unit 12 judges that the next video packet contains the sequence header code, the analyzing process operation is advanced to a step S47. To the contrary, when the stream analyzing unit 12 judges that the next video packet does not contain the sequence header code, the analyzing process operation is advanced to a step S40.

At the step S40, the stream analyzing unit 12 increments the pointer vpp of the video packet.

At the next step S41, the stream analyzing unit 12 checks as to whether or not the above-described I picture, or P picture is ended. When the P picture, or the I picture is not yet accomplished, the analyzing process operation is returned to the previous step S39. When the P picture, or the I picture is ended, the analyzing process operation is advanced to a step S42.
[0096] At this step S42, the stream analyzing unit 12 sets an address of a packet when the P picture, or the I picture is ended to "P_end_address" (see Fig. 10). With execution of the above-described analyzing process operation, the address of the first I picture, or P picture subsequent to the I picture is determined.

[0097] At a step S43, while video pointer vpp is not incremented, the stream analyzing unit 12 checks as to whether or not a next video packet contains a sequence header code. When the stream analyzing unit 12 judges that the next video packet contains the sequence header code, the analyzing process operation is advanced to the step S47. To the contrary, when the stream analyzing unit 12 judges that the next video packet does not contain the sequence header code, the analyzing process operation is advanced to a step S44.

[0098] At the step S44, the stream analyzing unit 12 increments the pointer vpp of the video packet.

[0099] At the next step S45, the stream analyzing unit 12 checks as to whether or not the P picture, or the I picture is ended. When the P picture, or the I picture is not yet accomplished, the analyzing process operation is returned to the step S43. When the P picture, or the I picture is ended, the analyzing process operation is advanced to a further step S46.

[0100] At this step S46, the stream analyzing unit 12 sets an address of a packet when the P picture, or the I picture is ended to "P_end_address" (see Fig. 10). With execution of the above-described analyzing process operation, the address of either the P picture, or the I picture located at two next positions of the I picture.

[0101] At a step S47, the stream analyzing unit 12 outputs the address of a packet when the P picture, or the I picture is ended to "P_end_address" (see Fig. 10). With execution of the above-described analyzing process operation, the address of either the P picture, or the I picture located at two next positions of the I picture.

[0102] At a step S48, the stream analyzing unit 12 judges as to whether or not the present packet corresponds to a finally entered packet. When the stream analyzing unit 12 judges that the present packet is not the finally entered packet, the analyzing process operation is returned to the step S33. To the contrary, when the present packet corresponds to the finally-inputted packet, the analyzing process operation is accomplished.

[0103] In such a case that a plurality of programs are contained in the transport stream to be recorded, the above-described analyzing operation of the video stream is carried out with respect to the video packet of the respective programs.

[0104] When the stream analyzing unit 12 produces the entry point data in accordance with the above-described manner, the stream analyzing unit 12 supplies this entry point data to the entry point map forming unit 16. The entry point map forming unit 16 processes the entry point data supplied from the stream analyzing unit 12 to form a table with respect to each of the programs, and then forms an entry point map as indicated in Fig. 9, which will be supplied to the file system unit 17.

[0105] As a result, the transport stream added with the time stamp by the time stamp adding unit 15, the time unit map and the entry point map, functioning as feature point data indicative of a feature point of this transport stream are supplied to the unit 17. The file system unit 17 processes the transport stream and the feature point data corresponding thereto as a file.

[0106] Fig. 31 represents an example of this file structure. In this file structural example, three sets of programs are multiplexed within the transport stream file. As shown in this drawing, the entry point map is constituted in such a manner that this entry point map follows the time unit map. Then, each of the entry point maps owns the below-mentioned data every program:

   (1) program_number of a program;
   (2) a PID of a transport packet of a PMT of a program;
   (3) both a PID and stream_type of a transport packet of video data which constitutes a program;
   (4) both a PID and stream_type of a transport packet of audio data which constitutes a program;
   (5) a PID of a PCR of a program; and
   (6) a list of an entry point.

[0107] The file produced by the file system unit 17 is supplied to the error correcting unit 18. After an error correction code is added to this produced file, the resulting file is supplied to the modulating unit 19 so as to be modulated by a preselected modulating system. A signal outputted from the modulating unit 19 is supplied to the writing unit 20, so that this supplied signal is written into the recording medium 21.

[0108] Both the transport stream and the feature point data thereof are recorded on the recording medium 21 in the above-described manner.

MODIFICATION OF MOVING PICTURE RECORDING APPARATUS

[0109] In the above-described embodiment, both the time unit map and the entry point map are formed from the
transport stream. Alternatively, for instance, in such a case that the moving picture recording apparatus itself multiplexes transport streams to produce a multiplexed transport stream, this moving picture recording apparatus may produce both a time unit map and an entry point map during this multiplexing operation. Fig. 32 indicates a structural example of this alternative case.

[0110] In other words in the structural example of Fig. 32, elementary streams #1 to #n of video data and elementary streams #1 to #n of audio data of a plurality ("n" pieces) of programs are inputted to a multiplexing unit 40. A system time clock unit 42 counts a system time clock having a frequency of 27 MHz, produces a time stamp, and then outputs the time stamp to both a controller 41 and a time unit map forming unit 43. The controller 41 analyzes the respective elementary streams which are inputted to the multiplexing unit 40, and controls this multiplexing unit 40 in such a manner that the multiplexing unit 40 multiplexes the transport streams while satisfying the T-STD (Transport Stream System Target Decoder) of the MPEG system rules.

[0111] The controller 41 supplies a packet number indicative of a total number of transport packets, which is derived from the multiplexing unit 40, to both a time unit map forming unit 43 and an entry point map forming unit 44. The time unit map forming unit 43 forms a time unit map based upon the packet number inputted from the controller 41 and the time stamp inputted from the system time clock 42.

[0112] Also, the controller 41 supplies program information and entry point data to the entry point map forming unit 44. The entry point map forming unit 44 forms an entry point map based upon the packet number, the program information, and the entry point data, which are supplied form the controller 41, and also the time unit map supplied from the time unit map forming unit 43.

[0113] The transport stream derived from the multiplexing unit 40, the time unit map formed by the time unit map forming unit 43, and the entry point map formed by the entry point map forming unit 44 are supplied to the file system unit 17 shown in Fig. 27, respectively. It should also be noted that the arrangement defined from this file system unit 17 to the recording medium 21 is similar to that shown in Fig. 27.

[0114] In the moving picture recording apparatus 1 with employment of the arrangement as shown in Fig. 32, the controller 41 produces both program information and entry point data from the elementary stream multiplexed by the multiplexing unit 40, and then supplies these program information and entry point data to the entry point map forming unit 44. Also, the controller 41 supplies the packet number to both the time unit map forming unit 43 and the entry point map forming unit 44. This packet number corresponds to the time stamp entered from the system time clock unit 42.

[0115] The time unit map forming unit 43 forms the time unit map based upon the packet number entered from the controller 41 and also the time stamp entered from the system time clock 42. Similarly, the entry point map forming unit 44 forms the entry point map based upon the packet number, the program information, and the entry point data, which are entered from the controller 41, and furthermore, the time unit map entered from the time unit map forming unit 43.

[0116] Similar to the arrangement shown in Fig. 27, the formed transport stream, time unit map, and entry point map are processed by the file system unit 17 as a file, and then, an error correction portion is added thereto by the error correcting unit 18. Then, the error-corrected transport stream/time unit map/entry point map are furthermore modulated by the modulating unit 19. Thereafter, the modulated transport stream/time unit map/entry point map are recorded on the recording medium 21 by the writing unit 20.

ARRANGEMENT OF MOVING PICTURE REPRODUCING APPARATUS

[0117] Next, a description will now be made of a moving picture reproducing apparatus for reproducing the recording medium 21 on which the transport stream file and the feature point data of this transport stream file have been recorded in the above-described manner. Fig. 33 indicates a structural example of such a moving picture reproducing apparatus 51. In this moving picture reproducing apparatus 51, a reading unit 61 reads the data recorded on the recording medium 21, and then outputs the read data to a demodulating unit 62. The demodulating unit 62 demodulates the data entered from the reading unit 61, and then outputs the demodulated data to an error correcting unit 63. The error correcting unit 63 corrects an error contained in the data inputted from the demodulating unit 62, and then supplies the error-corrected data to a file system unit 64.

[0118] The file system unit 64 separates the data entered from the error correcting unit 63 into both a transport stream file and feature point data, and then supplies this stream file to a buffer 65 and also this feature point data to a reproducing control unit 71. The reproducing control unit 71 controls the reading unit 61, a demultiplexer 69, and an AV decoder 70 in response to an instruction. This instruction is entered by a user via a terminal 73 and a user interface.

[0119] The buffer 65 supplies such a value saved from the stream file in arrival_time_stamp as an initial value to a counter 68 so as to set this counter 68. The counter 68 counts a clock having a frequency of 27 MHz, while using the initial value derived from the buffer 65 as the reference value, and then supplies the count value to a comparing unit 66. This clock is generated from the system time clock unit 67.

[0120] The comparing unit 66 compares the count value supplied from the counter 68 with the value arrival_time_stamp contained in the transport packet supplied from the buffer 65. When this count value is made coincident with the
value arrival_time_stamp, the comparing unit 66 outputs this transport packet to the demultiplexer 69.

[0121] The demultiplexer 69 extracts both video data and audio data of such a channel corresponding to an instruction issued from the reproducing control unit 71 from the transport stream file entered from the comparing unit 66, and then outputs both the extracted video data and the extracted audio data to an AV decoder 70. The AV decoder 70 decodes both the video data and the audio data, which are inputted from the demultiplexer 69, and then outputs the decoded video/audio data from a terminal 72.

REPRODUCING OPERATION OF MOVING PICTURE REPRODUCING APPARATUS

[0122] Subsequently, a reproducing operation of the moving picture reproducing apparatus will now be explained. Both the transport stream file and the feature point data of this transport stream recorded in the moving picture recording apparatus 1 shown in Fig. 27, or Fig. 32 are recorded on the recording medium 21. In this transport stream file, either one program or a plurality of programs are multiplexed.

[0123] First, the reproducing control unit 71 instructs the reading unit 61 to read out the feature pint data from the transport stream file. At this time, the reading unit 61 reads out the feature point data of the transport stream from the reading medium 21, and then outputs the read feature point data to the demodulating unit 62. The demodulating unit 62 demodulates the inputted feature point data, and then outputs the demodulated data to the error correcting unit 63. The error correcting unit 63 corrects the error of the entered data, and then supplies the error-corrected data to the file system unit 64. The file system unit 64 supplies the entered stream feature point data to the reproducing control unit 71.

[0124] The program number is inputted from the terminal 73 into the reproducing control unit 71. This program number is designated to be reproduced by the user interface. The reproducing control unit 71 reads out from the feature point data, a PID of a transport packet of a PMT of this designated program; a PID and stream-type of a transport packet of video data which constitutes this program; a PID and stream-type of a transport packet of audio data which constitutes this program; and a PID of a PCR. Thereafter, the reproducing control unit 71 outputs the read items to both the demultiplexer 69 and the AV decoder 70.

[0125] Furthermore, the reproducing control unit 71 instructs the reading unit 61 to read out the transport stream file. In response to this instruction, the reading unit 61 reads the transport stream file from the recording medium 21. Similar to the above-explained case, this data about the transport stream file is processed by the demodulating unit 62, the error correcting unit 63, and the file system unit 64 to the buffer 65.

[0126] The buffer 65 reads out the value stored in arrival_time_stamp from the inputted transport stream file, and supplies the read value as the initial value to the counter 68 so as to set this counter 68. Then, the counter 68 counts the clock generated from the system time clock unit 67 while using this initial value as the reference value, and then supplies this count value to the comparing unit 66. The comparing unit 66 reads out the value of arrival_time_stamp from the transport stream file supplied from the buffer 65, and then compares this read value with the counter value supplied from the counter 68. At such timing when the read data is made coincident with the count value, the comparing unit 66 outputs the transport stream file to the demultiplexer 69.

[0127] The demultiplexer 69 separates the transport packet of the video data and the transport packet of the audio data, which constitute the program designated by the user interface, from the inputted transport stream. Then, the demultiplexer 69 enters these separated transport packets of the video/audio data into the AV decoder 70. The AV decoder 70 decodes both the video stream and the audio stream, and then outputs these decoded streams as a reproduced video signal and a reproduced audio signal from the terminal 72.

[0128] When the random access reproducing operation is instructed via the user interface, the reproducing control unit 71 determines a data reading position with respect to the recording medium 21 in response to the content of the feature point data of the stream stored therein, and then enters random access control information to the reading unit 61. For example, in the case that a program selected by a user is reproduced from a preset time instant, the reproducing control unit 71 calculates an address of a transport stream corresponding to a designated time instant, and then instructs the reading unit 61 to read the data at this calculated address from the recording medium 21. Next, this data reading sequential operation is described as follows:

[0129] A first explanation is made of the time unit map indicated in Fig. 7. Assuming now that a time instant of head data of a zeroth time unit TU0 is equal to "start_time", a time instant of head data of an N-th time unit (N > 0) becomes:

\[
\text{start_time} + \text{first_time_unit_size} + (N - 1) \times \text{time_unit_size},
\]

As a result, the following fact can be revealed. That is, if the reproducing control unit 71 can grasp such a number of a time unit in which a time instant of head data of a time unit becomes larger than a time instant designated by the user, then the reading unit 61 may read data from the time unit having this grasped number.

[0130] In this case, assuming now that an address of the head data of the zeroth time unit on the recorded stream is equal to 0, an address "time_unit_address (N)" of the head data of the N-th time unit may be calculated by the below-mentioned formula (1):
Equation 1

\[
\text{time}_\text{unit}_\text{address}(N) = \sum_{i=0}^{N} \text{delta}_\text{time}_\text{unit}_\text{address}(i)
\]
64, the buffer 65, and the comparing unit 66. The demultiplexed data is decoded by the AV decoder 70, and then the AV decoder 70 outputs the decoded data.

The above-explained calculation process operation defined at this step S63 will now be described more in detail in the case of the time unit map shown in Fig. 7 with reference to flow charts indicated in Figs. 35 and 36. At a step S81 of the flow chart shown in Fig. 35, when both program-number and a reproduction starting time instant "Tst" are entered from the terminal 73 to the reproducing control unit 71, the reproducing control unit 71 judges as to whether or not the reproducing starting time instant Tst entered at the step S81 is equal to the starting time instant "start_time" (Fig. 3B) of the transport stream contained in the feature point data at a step S82. When the reproduction starting time instant Tst is equal to the starting time "start_time", the process operation is advanced to a step S86. At this step S86, the reproducing control unit 71 sets "0" to a variable "N" indicative of a number of a time unit, and also sets "0" to an address "time_unit_address (N)" of this time unit (namely, zeroth time unit).

To the contrary, when the reproducing control unit 71 judges that the reproduction starting time instant Tst is not equal to the starting time instant "start_time", at the step S82, the process operation is advanced to a further step S83. At this step S83, the reproducing control unit 71 reads a header portion of the time unit map, and then calculates such a minimum value "N" capable of satisfying the below-mentioned inequality:

\[ Tst \leq start_time + first_time_unit_size + (N-1) \times time_unit_size \]

At a step S85, the reproducing control unit 71 calculates time_unit_address (N) based on the data of the time unit map in accordance with the formula (1).

When a calculation is made of a time instant "time_unit_address (N)" of the head data in the N-th time unit, the reproducing control unit 71 instructs the reading unit 61 to read data from the address "time_unit_address (N)" of the N-th time unit at step S87.

In response to an instruction issued from the reproducing control unit 71, the reading unit 61 reads out a transport stream from this address "time_unit_address (N)" from the recording medium 21 at a step S88. The data (transport stream) read by this reading unit 61 is supplied via the demodulating unit 62, the error correcting unit 63, the file system unit 64, the buffer 65, and the comparing unit 66 to the demultiplexer 69.

At a step S89, the reproducing control unit 71 outputs to the demultiplexer 69, program_number of such a program to which the user instructs to reproduce. At a step S90, the demultiplexer 69 separates a transport packet of the program at program_number instructed by the reproducing control unit 71, and then outputs this separated transport packet to the AV decoder 70. At a step S91, the AV decoder 70 decodes the data entered from the demultiplexer 69, and then outputs the decoded data from the terminal 72.

Referring now to a flow chart shown in Fig. 37, the calculation process operation defined at the step S63 of the flow chart shown in Fig. 34 will be described more in detail in the case of the time unit map shown in Fig. 22D.

The process steps of the flow chart indicated in Fig. 37 are made in such a manner that the step S82, S84, S85 defined in the flow charts of Fig. 35 and Fig. 36 are changed into steps S102, S104, and S105, respectively. Other steps than the above-explained steps defined in the flow chart of Fig. 37 are equal to the steps defined in the above-explained flow charts of Fig. 35 and Fig. 36. Only the respective changed steps S102, S104, and S105 will now be explained as follows:

At the step S102, the reproduction starting time instant Tst is compared with a starting time "offset_time" of a time unit map.

At a step S104, a calculation is made of a minimum value "N" capable of satisfying the below-mentioned inequality:

\[ Tst \leq offset_time + N \times time_unit_size \]

At the step S85, the reproducing control unit 71 calculates an address "time_unit_address (N)" based upon the data of the time unit map in accordance with the below-mentioned formula:
It should be understood that a series of the above-described process operations may be executed by using hardware, but also may be executed by utilizing software. In the case that a series of the above-explained process operations is executed by employing software, the following cases may be conceived. That is, such a computer may be employed in which a program for constituting this software is installed in a moving picture recording/reproducing apparatus functioning as exclusively-used hardware, and a general-purpose computer capable of executing various sorts of functions may be employed, while various sorts of programs are installed.

Referring now to Fig. 38, a recording medium according to an embodiment mode of the present invention will be explained. That is to say, while a program capable of executing a series of the above-described process operations is installed in a general-purpose personal computer, this computer program can be recorded on such a recording medium of the embodiment mode.

As indicated in Fig. 38A, the computer program may be provided under such a condition that this computer program is previously installed in either a hard disk drive (HDD) 302 or a semiconductor memory 303, functioning as a recording medium built in a computer 301.

Alternatively, as represented in Fig. 38B, the computer program may be provided as package software in such a manner that this computer program is temporarily stored, or permanently stored into a recording medium such as a floppy disk 311, a CD-ROM (Compact Disk-Read Only Memory) disk 312, an MO (Magneto-Optical) disk 313, and DVD (Digital Versatile Disk) 314, a magnetic disk 315, and a semiconductor memory 316.

Furthermore, as shown in Fig. 38C, the computer program may be transmitted from a download site 321 via a satellite 322 used in a digital satellite broadcasting system to a computer 323 in a wireless manner. Alternatively, the computer program may be transmitted from the download site 321 via a network 131 such as a local area network (LAN) and the Internet to the computer 323 in a wired line manner, and then may be stored into a hard disk drive built in this computer 323.

A recording medium defined in this specification implies such a wide conceptional idea involving all of the above-explained recording media.

It should also be noted that the steps used to describing the computer program provided from the reading medium may be apparently processed in the time sequential manner in accordance with the described sequential order, but may not be time sequentially processed, for example, may be processed in a parallel manner, or a separate manner.

It should also be understood that the system described in this specification represents an entire apparatus arranged by a plurality of apparatuses.

As previously explained, in the case that the recording medium on which either one transport stream or a plurality of transport streams are recorded is random-accessed so as to reproduce desirable data, since the starting position of the I picture, or the audio frame can be effectively searched, the high-speed random access reproducing operation can be carried out in response to the user input.

As previous explained in detail, in accordance with the first data processing apparatus, data processing method, and the program recorded on the recording medium, the coded stream is segmented with respect to a predetermined time unit, and then such a time unit map is formed. This time unit map indicates the address of the data with respect to the time unit of the segmented coded stream. As a result, the random access operation can be carried out in high speed response.

Furthermore, in accordance with the second data processing apparatus, data processing method, and the program recorded on the recording medium, the coded stream recorded on the recording medium is reproduced from an arbitrary position based upon the time unit map reproduced from the recording medium. As a consequence, the random data stream production can be carried out in a high speed.

Claims

1. A data processing apparatus for processing data contained in an entered coded stream comprising transport packets, the apparatus being characterised by:

   a time stamp generator (14) operable to generate time stamps each indicative of an arrival time of an associated one of the transport packets of the entered coded stream;

   a source packet generator (15) operable to generate source packets by adding the arrival time stamps to the
associated transport packets; and
a time unit map generator (23) responsive to said time stamps to generate a time unit map that comprises addresses \( A(TUi) \) of each of a plurality of successive time units \( (TUi) \) of like duration into which the entered coded stream is segmented.

2. Apparatus as claimed in claim 1, wherein the time unit map generator (23) is operable to include, in the time unit map, information indicative of the amount of data in each time unit \( (TUi) \).

3. Apparatus as claimed in claim 2, wherein the time unit map generator (23) is operable to generate a time unit map that comprises the address \( A(TUi) \) of a head packet of each said time unit \( (TUi) \), and an interval \( (e.g. A(TU1) - A(TU0)) \) between addresses corresponding to the amount of data in each time unit.

4. Apparatus as claimed in claim 2, wherein the time unit map generator (23) is operable to include, in the time unit map, offset information indicative of offset values of the source packets, and addresses indicating a relative number of the source packet with respect to the offset value.

5. Apparatus as claimed in any one of the preceding claims, comprising means (24) for producing packet numbers, the time unit map generator (23) being responsive to said packet numbers, as well as said time stamps, to generate the time unit map.

6. Apparatus as claimed in any one of the preceding claims, for processing data contained in an entered coded stream comprising transport packets that relate to respective ones of a plurality of programs, wherein:

   - selection means (11) is provided for selecting the transport packets that relate to a desired program from the encoded stream; and
   - the source packet generator (15) is operable to generate source packets only for the selected transport packets.

7. Apparatus as claimed in claim 6, wherein
   - the selection means (11) is capable of selecting the transport packets that relate to plural desired programs from the encoded stream; and
   - the time unit map generator (23) is operative to generate a respective said time unit map for each of the desired programs.

8. Apparatus as claimed in any one of the preceding claims, wherein the time unit map generator is operable to change the content of said time unit map when said coded stream is edited.

9. Apparatus as claimed in any one of the preceding claims, comprising an entry point map forming generator (16) for generating an entry point map subordinated to said time unit map, said entry point map indicating a position of an entry point of said coded stream.

10. Apparatus as claimed in claim 9, wherein the entry point forming generator (16) is so operative that said entry point map includes an interval of addresses defined from head data of said time unit of the entry point.

11. Apparatus as claimed in claim 10, wherein the entry point map forming generator (16) is so operative that:

    - said entry point represents a head address of an I picture contained in said time unit; and
    - said entry point map contains an interval defined from said head address of the I picture up to an end address of said I picture, or an interval of addresses contained within a time unit including said I picture.

12. Apparatus as claimed in claim 9, claim 10 or claim 11, wherein the entry point map forming generator (16) is so operative that said entry point map includes a flag for indicating as to whether or not said entry point within said time unit is present.

13. Apparatus as claimed in any one of claims 9 to 12, when appendant to claim 7, wherein the entry point map forming generator (16) is operative to generate a respective said entry point map for each of the desired programs.

14. Apparatus as claimed in any one of claims 9 to 13, wherein the entry point map forming generator is operative to
change said entry point map when said coded stream is edited.

15. Apparatus as claimed in any one of the preceding claims, comprising filing means (17) for filing the time unit map in connection with the generated source packets.

16. Apparatus as claimed in any one of the preceding claims, comprising recording means (20) operable to record said source packets, along with said time unit map, on a recording medium (21).

17. A data processing method for processing data contained in an entered coded stream comprising transport packets, the method being characterised by the steps of:

   generating (14) time stamps each indicative of an arrival time of an associated one of the transport packets of the entered coded stream;
   generating (15) source packets by adding the arrival time stamps to the associated transport packets; and
   in response to said time stamps, generating (23) a time unit map that comprises addresses (A(TUi)) of each of a plurality of successive time units (TUi) of like duration into which the entered coded stream is segmented.

18. A method as claimed in claim 17, wherein the time unit map generating step (23) includes, in the time unit map, information indicative of the amount of data in each time unit (TUi).

19. A method as claimed in claim 18, wherein the time unit map generation step (23) generates a time unit map that comprises the address (A(TUi)) of a head packet of each said time unit (TUi), and an interval (e.g. A(TU1) - A(TU0)) between addresses corresponding to the amount of data in each time unit.

20. A method as claimed in claim 18, wherein the time unit map generation step (23) includes, in the time unit map, offset information indicative of offset values of the source packets, and addresses indicating a relative number of the source packet with respect to the offset value.

21. A method as claimed in any one of claims 17 to 20, comprising a step (24) of producing packet numbers, the time unit map generation step (23) being responsive to said packet numbers, as well as said time stamps, to generate the time unit map.

22. A method as claimed in any one of claims 17 to 21, for processing data contained in an entered coded stream comprising transport packets that relate to respective ones of a plurality of programs, wherein the transport packets that relate to a desired program are selected (11) from the encoded stream, and the source packet generation step (15) comprises generating source packets only for the selected transport packets.

23. A method as claimed in claim 22, wherein the transport packets that relate to plural desired programs are selected (11) from the encoded stream, and a respective said time unit map is generated (23) for each of the desired programs.

24. A method as claimed in any one of claims 17 to 23, wherein the content of said time unit map is changed when said coded stream is edited.

25. A method as claimed in any one of claims 17 to 24, comprising generating (16) an entry point map subordinated to said time unit map, said entry point map indicating a position of an entry point of said coded stream.

26. A method as claimed in claim 25, wherein said entry point map includes an interval of addresses defined from head data of said time unit of the entry point.

27. A method as claimed in claim 26, wherein:

   said entry point represents a head address of an I picture contained in said time unit; and
   said entry point map contains either an interval defined from said head address of the I picture up to an end address of said I picture, or an interval defined from said head address of the I picture up to an end address of a P picture subsequent to said end address of the I picture, contained within a time unit including said I picture.

28. A method as claimed in claim 25, claim 26 or claim 27, wherein said entry point map includes a flag for indicating as to whether or not said entry point within said time unit is present.
29. A method as claimed in any one of claims 25 to 28, when appendant to claim 23, comprising generating a respective said entry point map for each of the desired programs.

30. A method as claimed in any one of claims 25 to 29, wherein said entry point map is changed when said coded stream is edited.

31. A method as claimed in any one of claims 17 to 30, comprising filing (17) the time unit map in connection with the generated source packets.

32. A method as claimed in any one of claims 17 to 31, comprising recording said source packets, along with said time unit map, on a recording medium (21).

33. A recording medium having recorded thereon a computer readable program for processing data contained in an entered coded stream comprising transport packets, the program comprising the steps of any one of claims 17 to 32.

34. Apparatus for reproducing, from a recording medium, source packets and a time unit map recorded on the medium by a method as claimed in claim 32, the apparatus comprising means (61-71) operable to reproduce the time unit map from the recording medium and then to reproduce the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map.

35. Apparatus as claimed in claim 34 for reproducing, from a recording medium, source packets, a time unit map and an entry point map recorded on the medium by a method as claimed in claim 32 when appendant to any one of claims 25 to 28, the apparatus comprising means (61-71) operable to reproduce the time unit map and the entry point map from the recording medium and then to reproduce the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map and entry point map.

36. A method of reproducing, from a recording medium, source packets and a time unit map recorded on the medium by a method as claimed in claim 32, the reproducing method comprising the steps of reproducing the time unit map from the recording medium and then reproducing the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map.

37. A method as claimed in claim 36, for reproducing, from a recording medium, source packets, a time unit map and an entry point map recorded on the medium by a method as claimed in claim 32 when appendant to any one of claims 25 to 28, the reproducing method comprising reproducing the time unit map and the entry point map from the recording medium and then reproducing the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map and entry point map.

38. A recording medium having recorded thereon a computer readable program for reproducing, from a recording medium, source packets and a time unit map recorded on the medium by a method as claimed in claim 32, the program comprising the steps of reproducing the time unit map from the recording medium and then reproducing the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map.

39. A recording medium having recorded thereon a computer readable program for reproducing, from a recording medium, source packets, a time unit map and an entry point map recorded on the medium by a method as claimed in claim 32 when appendant to any one of claims 25 to 28, the program comprising the steps of reproducing the time unit map and the entry point map from the recording medium and then reproducing the source packets from an arbitrary position on the recording medium based upon the reproduced time unit map and entry point map.

40. A recording medium having recorded thereon:

source packets generated by adding, to transport packets of a coded data stream, arrival time stamps each indicative of an arrival time of an associated one of the transport packets of the coded stream; and

a time unit map that comprises addresses (A(TUi)) of each of a plurality of successive time units (TUi) of like duration into which the coded stream is segmented.

41. A recording medium as claimed in claim 40, wherein the time unit map includes information indicative of the amount of data in each time unit (TUi).
42. A recording medium as claimed in claim 41, wherein the time unit map comprises the address (A(TUi)) of a head packet of each said time unit (TU), and an interval (e.g. A(TU1) - A(TU0)) between addresses corresponding to the amount of data in each time unit.

43. A recording medium as claimed in claim 41, wherein the time unit map includes offset information indicative of offset values of the source packets, and addresses indicating a relative number of the source packet with respect to the offset value.

44. A recording medium as claimed in any one of claims 40 to 43, wherein the time unit map comprises information derived from packet numbers, as well as said time stamps.

45. A recording medium as claimed in any one of claims 40 to 44, wherein the source packets have been generated only for selected ones of the transport packets of the coded stream, which relate to respective ones of a plurality of programs, the selected transport packets relating to a desired one of the plurality of programs.

46. A recording medium as claimed in any one of claims 40 to 44, wherein the source packets have been generated only for selected ones of the transport packets of the coded stream, which relate to respective ones of a plurality of programs, the selected transport packets relating to a desired plural number of the plurality of programs.

47. A recording medium as claimed in any one of claims 40 to 46, that has recorded thereon an entry point map subordinated to said time unit map, said entry point map indicating a position of an entry point of said coded stream.

48. A recording medium as claimed in claim 47, wherein said entry point map includes an interval of addresses defined from head data of said time unit of the entry point.

49. A recording medium as claimed in claim 48, wherein:

   said entry point represents a head address of an I picture contained in said time unit; and
   said entry point map contains either an interval defined from said head address of the I picture up to an end address of said I picture, or an interval defined from said head address of the I picture up to an end address of a P picture subsequent to said end address of the I picture, contained within a time unit including said I picture.

50. A recording medium as claimed in claim 47, claim 48 or claim 49, wherein said entry point map includes a flag for indicating as to whether or not said entry point within said time unit is present.

51. A recording medium as claimed in any one of claims 47 to 50, when appendant to claim 46, comprising a respective said entry point map for each of the desired programs.

Patentansprüche

1. Datenverarbeitungsgerät zur Verarbeitung von Daten, die in einem eingegebenen codierten Strom enthaltenen sind, der Transportpakete aufweist, wobei das Gerät **gekennzeichnet ist durch**:

   einen Zeitstempelgenerator (14), der zum Erzeugen von Zeitstempeln betriebsfähig ist, deren jeder eine Ankunftszeit eines zugeordneten der Transportpakete des eingegebenen codierten Stroms anzeigt,
   einen Quellenpaketgenerator (15), der betriebssfähig ist, um Quellenpakete durch Hinzufügen der Ankunftszeitstempel zu den zugeordneten Transportpaketen zu erzeugen, und
einen Zeiteinheitsabbildungsgenerator (23), der auf die Zeitstempel reagiert, um eine Zeiteinheitsabbildung zu erzeugen, die Adressen (A(TUi)) jeder von mehreren sukzessiven Zeiteinheiten (TUi) ähnlicher Dauer, in die der eingegebene codierte Strom segmentiert ist, aufweist.

2. Gerät nach Anspruch 1, wobei der Zeiteinheitsabbildungsgenerator (23) betriebssfähig ist, um in der Zeiteinheitsabbildung Information zu inkludieren, welche die Menge von Daten in jeder Zeiteinheit (TUi) anzeigt.

3. Gerät nach Anspruch 2, wobei der Zeiteinheitsabbildungsgenerator (23) betriebssfähig ist, um eine Zeiteinheitsabbildung, welche die Adresse (A(TUi)) eines Kopfpakets jeder Zeiteinheit (TUi) aufweist, und ein mit der Menge von Daten in jeder Zeiteinheit korrespondierendes Intervall (beispielsweise A(TU1) - A(TU0)) zwischen Adressen zu er-
zeugen.

4. Gerät nach Anspruch 2, wobei der Zeiteinheitsabbildungsgenerator (23) betriebsfähig ist, um in der Zeiteinheitsabbildung eine Verschiebeinformation, die Verschiebewerte der Quellenpakete anzeigt, und Adressen, die eine relative Zahl des Quellenpakets in Bezug auf den Verschiebewert anzeigen, zu inkludieren.

5. Gerät nach einem der vorhergehenden Ansprüche, mit einer Einrichtung (24) zur Erzeugung von Paketzahlen, wobei der Zeiteinheitsabbildungsgenerator (23) auf die Paketzahlen sowie die Zeitstempel reagiert, um eine Zeiteinheitsabbildung zu erzeugen.

6. Gerät nach einem der vorhergehenden Ansprüche, zur Verarbeitung von Daten, die in einem eingegebenen codierten Strom enthalten sind, der Transportpakete aufweist, die sich auf jeweilige einzelne von mehreren Programmen beziehen, wobei:

   eine Auswähleinrichtung (11) zum Auswählen der Transportpakete aus dem codierten Strom, die sich auf ein gewünschtes Programm beziehen, vorhanden ist und der Quellenpaketgeneratoren (15) betriebsfähig ist, um Quellenpakete nur für die ausgewählten Transportpakete zu erzeugen.

7. Gerät nach Anspruch 6, wobei die Auswähleinrichtung (11) fähig ist zum Auswählen der Transportpakete aus dem codierten Strom, die sich auf mehrere gewünschte Programme beziehen, und der Zeiteinheitsabbildungsgenerator (23) betriebsfähig ist, um für jedes der gewünschten Programme eine jeweilige Zeiteinheitsabbildung zu erzeugen.

8. Gerät nach einem der vorhergehenden Ansprüche, wobei der Zeiteinheitsabbildungsgenerator betriebsfähig ist, um den Inhalt der Zeiteinheitsabbildung zu ändern, wenn der codierte Strom editiert wird.


10. Gerät nach Anspruch 9, wobei der Eintrittspunktabbildungsgenerator (16) so betriebsfähig ist, dass die Eintrittspunktabbildung ein Intervall von Adressen aufweist, das von Kopfdaten der Zeiteinheit des Eintrittspunkts definiert ist.


12. Gerät nach Anspruch 9, Anspruch 10 oder Anspruch 11, wobei der Eintrittspunktabbildungs-Bildungsgenerator (16) so betriebsfähig ist, dass die Eintrittspunktabbildung ein Kennzeichen zum Anzeigen, ob der Eintrittspunkt in der Zeiteinheit präsent ist oder nicht, aufweist.


14. Gerät nach einem der Ansprüche 9 bis 13, wobei der Eintrittspunktabbildungs-Bildungsgenerator betriebsfähig ist, um die Eintrittspunktabbildung zu ändern, wenn der codierte Strom editiert wird.


16. Gerät nach einem der vorhergehenden Ansprüche, mit einer Aufzeichnungseinrichtung (20), die betriebsfähig ist, um die Quellenpakete zusammen mit der Zeiteinheitsabbildung auf einem Aufzeichnungsmedium (21) aufzuzeichnen.
17. Datenverarbeitungsverfahren zur Verarbeitung von Daten, die in einem eingegebenen codierten Strom enthalten sind, der Transportpakete aufweist, wobei das Verfahren gekennzeichnet ist durch die Schritte:

   Erzeugen (14) von Zeitstempeln, deren jeder eine Ankunftszeit eines zugeordneten der Transportpakete des eingegebenen codierten Stroms anzeigt,
   Erzeugen (15) von Quellenpaketen durch Hinzufügen der Ankunftszeitstempel zu den zugeordneten Transportpaketen, und
   Erzeugen (23) in Reaktion auf die Zeitstempel einer Zeiteinheitsabbildung, die Adressen \((A(T_{Ui}))\) jeder von mehreren sukzessiven Zeiteinheiten \((T_{Ui})\) ähnlicher Dauer, in die der eingegebene codierte Strom segmentiert ist, aufweist.

18. Verfahren nach Anspruch 17, wobei der Zeiteinheitsabbildungs-Erzeugungsschritt (23) in der Zeiteinheitsabbildung Information aufweist, welche die Menge von Daten in jeder Zeiteinheit \((T_{Ui})\) anzeigt.

19. Verfahren nach Anspruch 18, wobei der Zeiteinheitsabbildungs-Erzeugungsschritt (23) eine Zeiteinheitsabbildung, welche die Adresse \((A(T_{Ui}))\) eines Kopfpackets jeder Zeiteinheit \((T_{Ui})\) aufweist, und ein mit der Menge von Daten in jeder Zeiteinheit korrespondierendes Intervall (beispielsweise \((A(T_{U1}) - A(T_{U0}))\) zwischen Adressen erzeugt.

20. Verfahren nach Anspruch 18, wobei der Zeiteinheitsabbildungs-Erzeugungsschritt (23) in der Zeiteinheitsabbildung Verschiebeinformation, die Verschiebewerte der Quellenpakete anzeigt, und Adressen, die eine relative Zahl des Quellenpakets in Bezug auf den Verschiebewert anzeigt, aufweist.


22. Verfahren nach einem der Ansprüche 17 bis 21, zur Verarbeitung von Daten, die in einem eingegebenen codierten Strom enthalten sind, der Transportpakete aufweist, die sich auf jeweilige einzelne von mehreren Programmen beziehen, wobei die Transportpakete, die sich auf ein gewünschtes Programm beziehen, vom codierten Strom ausgewählt (11) werden, und wobei der Quellenpaketerzeugungsschritt (15) die Erzeugung von Quellenpaketen nur für die ausgewählten Transportpakete aufweist.

23. Verfahren nach Anspruch 22, wobei die Transportpakete, die sich auf mehrere gewünschte Programme beziehen, vom codierten Strom ausgewählt (11) werden und für jedes der gewünschten Programme je eine Zeiteinheitsabbildung erzeugt (23) wird.

24. Verfahren nach einem der Ansprüche 17 bis 23, wobei der Inhalt der Zeiteinheitsabbildung geändert wird, wenn der codierte Strom editiert wird.


27. Verfahren nach Anspruch 26, wobei:


28. Verfahren nach Anspruch 25, Anspruch 26 oder Anspruch 27, wobei die Eintrittspunktabbildung ein Kennzeichen zum Anzeigen, ob der Eintrittspunkt in der Zeiteinheit präsent ist oder nicht, aufweist.

30. Verfahren nach einem der Ansprüche 25 bis 29, wobei die Eintrittspunktabbildung geändert wird, wenn der codierte Strom editiert wird.


32. Verfahren nach einem der Ansprüche 17 bis 31, mit einem Aufzeichnen der Quellenpakete zusammen mit der Zeiteinheitsabbildung auf einem Aufzeichnungsmedium (21).

33. Aufzeichnungsmedium mit einem darauf aufgezeichneten computerlesbaren Programm zur Verarbeitung von Daten, die in einem eingegebenen codierten Strom enthalten sind, der Transportpakete aufweist, wobei das Programm die Schritte eines der Ansprüche 17 bis 32 aufweist.

34. Gerät zur Wiedergabe von Quellenpaketen und einer Zeiteinheitsabbildung von einem Aufzeichnungsmedium, die durch ein Verfahren nach Anspruch 32 auf dem Aufzeichnungsmedium aufgezeichnet sind, wobei das Gerät eine Einrichtung (61-71) aufweist, die betriebsfähig ist, um die Zeiteinheitsabbildung vom Aufzeichnungsmedium wiederzugeben und dann die Quellenpakete auf Basis der wiedergegebenen Zeiteinheitsabbildung von einer beliebigen Position auf dem Aufzeichnungsmedium wiederzugeben.

35. Gerät nach Anspruch 34 zur Wiedergabe von Quellenpaketen, einer Zeiteinheitsabbildung und einer Eintrittspunktabbildung vom Aufzeichnungsmedium von einem Aufzeichnungsmedium, die auf dem Aufzeichnungsmedium durch ein Verfahren nach Anspruch 32 und einem der Ansprüche 25 bis 28 aufgezeichnet sind, wobei das Gerät eine Einrichtung (61-71) aufweist, die betriebsfähig ist, um die Zeiteinheitsabbildung und die Eintrittspunktabbildung vom Aufzeichnungsmedium wiederzugeben und dann die Quellenpakete auf Basis der wiedergegebenen Zeiteinheitsabbildung von einer beliebigen Position auf dem Aufzeichnungsmedium wiederzugeben.


38. Aufzeichnungsmedium mit einem darauf aufgezeichneten computerlesbaren Programm zur Wiedergabe von Quellenpaketen und einer Zeiteinheitsabbildung von einem Aufzeichnungsmedium, die auf dem Aufzeichnungsmedium durch ein Verfahren nach Anspruch 32 aufgezeichnet sind, wobei das Programm die Schritte einer Wiedergabe der Zeiteinheitsabbildung vom Aufzeichnungsmedium und dann Wiedergabe der Quellenpakete auf Basis der wiedergegebenen Zeiteinheitsabbildung von einer beliebigen Position auf dem Aufzeichnungsmedium aufweist.


40. Aufzeichnungsmedium, auf dem aufgezeichnet sind:
Quellenpakete, die durch Hinzufügen zu Transportpaketen eines codierten Datenstroms von Ankunftszeitstempeln, deren jeder eine Ankunftszeit eines zugeordneten der Transportpakete des codierten Stroms anzeigt, erzeugt werden, und

eine Zeiteinheitsabbildung, die Adressen (A(TUi)) jeder von mehreren sukzessiven Zeiteinheiten (TU_i) gleicher Dauer, in die der codierte Datenstrom segmentiert ist, aufweist.

41. Aufzeichnungsmedium nach Anspruch 40, wobei die Zeiteinheitsabbildung Information aufweist, welche die Menge von Daten in jeder Zeiteinheit (TU_i) anzeigt.

42. Aufzeichnungsmedium nach Anspruch 41, wobei die Zeiteinheitsabbildung die Adresse (A(TUi)) eines Kopfpakets jeder Zeiteinheit (TU_i) und ein mit der Menge von Daten in jeder Zeiteinheit korrespondierendes Intervall (beispielsweise A(TU_1)-A(TU_0)) zwischen Adressen aufweist.

43. Aufzeichnungsmedium nach Anspruch 41, wobei die Zeiteinheitsabbildung eine Verschiebeinformation, die Verschiebewerte der Quellenpakete anzeigt, und Adressen, die eine relative Zahl des Quellenpakets in Bezug auf den Verschiebewert anzeigen, aufweist.

44. Aufzeichnungsmedium nach einem der Ansprüche 40 bis 42, wobei die Zeiteinheitsabbildung von Paketzahlen abgeleitete Information sowie die Zeitstempel aufweist.

45. Aufzeichnungsmedium nach einem der Ansprüche 40 bis 44, wobei die Quellenpakete nur für ausgewählte der Transportpakete des codierten Stroms, die sich auf jeweilige von mehreren Programmen beziehen, erzeugt worden sind, wobei sich die ausgewählten Transportpakete auf gewünschte der mehreren Programme beziehen.

46. Aufzeichnungsmedium nach einem der Ansprüche 40 bis 44, wobei die Quellenpakete nur für ausgewählte der Transportpakete des codierten Datenstroms, die sich auf jeweilige von mehreren Programmen beziehen, erzeugt worden sind, wobei sich die ausgewählten Transportpakete auf eine gewünschte Mehrzahl der mehreren Programme beziehen.

47. Aufzeichnungsmedium nach einem der Ansprüche 40 bis 46, das eine darauf aufgezeichnete Eintrittspunktabbildung aufweist, die der Zeiteinheitsabbildung untergeordnet ist, wobei die Eintrittspunktabbildung eine Position eines Eintrittspunkts des codierten Stroms anzeigt.


50. Aufzeichnungsmedium nach Anspruch 47, Anspruch 48 oder Anspruch 49, wobei der Eintrittspunktabbildung ein Kennzeichen zur Anzeige, ob der Eintrittspunkt in der Zeiteinheit präsent ist oder nicht, aufweist.

51. Aufzeichnungsmedium nach einem der Ansprüche 47 bis 50 und Anspruch 46, mit jeweils einer Eintrittspunktabbildung für jedes der gewünschten Programme.

Revendications

1. Appareil de traitement de données pour traiter des données qui sont contenues dans un train codé entré comprenant des paquets de transport, l’appareil étant caractérisé par :

   un génératore de timbre temporel (14) qui peut fonctionner pour générer des timbres temporels dont chacun est indicatif d’un temps d’arrivée de l’un associé des paquets de transport du train codé entré ;

   un génératore de paquet de source (15) qui peut fonctionner pour générer des paquets de source en additionnant
les timbres temporels d’arrivée aux paquets de transport associés ; et
un générateur de carte d’unités temporelles (23) qui est sensible auxdits timbres temporels pour générer une
carte d’unités temporelles qui comprend des adresses (A(TU)) de chacune d’une pluralité d’unités temporelles
(TU) successives de durées identiques selon lesquelles le train codé entré est segmenté.

2. Appareil selon la revendication 1, dans lequel le générateur de carte d’unités temporelles (23) peut fonctionner de
maniè re à inclure, dans la carte d’unités temporelles, une information qui est indicative de la quantité de données
dans chaque unité temporelle (TU).

3. Appareil selon la revendication 2, dans lequel le générateur de carte d’unités temporelles (23) peut fonctionner de
maniè re à générer une carte d’unités temporelles qui comprend l’adresse (A(TU)) d’un paquet de tête de chaque
dite unité temporelle (TU) et un intervalle (par exemple A(TU1)-A(TU0)) entre des adresses correspondant à la
quantité de données dans chaque unité temporelle.

4. Appareil selon la revendication 2, dans lequel le générateur de carte d’unités temporelles (23) peut fonctionner de
maniè re à inclure, dans la carte d’unités temporelles, une information qui est indicative de la quantité de données
dans chaque unité temporelle.

5. Appareil selon l’une quelconque des revendications précédentes, comprenant un moyen (24) pour produire des
numéros de paquet, le générateur de carte d’unités temporelles (23) étant sensible auxdits numéros de paquet de
même qu’auxdits timbres temporels pour générer la carte d’unités temporelles.

6. Appareil selon l’une quelconque des revendications précédentes, pour traiter des données qui sont contenues dans
un train codé entré comprenant des paquets de transport qui concernent ceux respectifs d’une pluralité de program-
mes, dans lequel :

   un moyen de sélection (11) est prévu pour sélectionner les paquets de transport qui concernent un programme
   souhaité à partir du train codé ; et
   le générateur de paquet de source (15) peut fonctionner pour générer des paquets de source seulement pour
   les paquets de transport sélectionnés.

7. Appareil selon la revendication 6, dans lequel :

   le moyen de sélection (11) peut sélectionner les paquets de transport qui concernent plusieurs programmes
   souhaités à partir du train codé ; et
   le générateur de carte d’unités temporelles (23) fonctionne pour générer une dite carte d’unités temporelles
   respective pour chacun des programmes souhaités.

8. Appareil selon l’une quelconque des revendications précédentes, dans lequel le générateur de carte d’unités tem-
   porelles peut fonctionner pour modifier le contenu de ladite carte d’unités temporelles lorsque ledit train codé est édité.

9. Appareil selon l’une quelconque des revendications précédentes, comprenant un générateur de formation de carte
de points d’entrée (16) pour générer une carte de points d’entrée qui est subordonnée à ladite carte d’unités tem-
porelles, ladite carte de points d’entrée indiquant une position d’un point d’entrée dudit train codé.

10. Appareil selon la revendication 9, dans lequel le générateur de formation de points d’entrée (16) fonctionne de telle
    sorte que ladite carte de points d’entrée inclue un intervalle d’adresses qui est défini à partir de données de tête de
    ladite unité temporelle du point d’entrée.

11. Appareil selon la revendication 10, dans lequel le générateur de formation de carte de points d’entrée (16) fonctionne
de telle sorte que :

    ledit point d’entrée représente une adresse de tête d’une image I qui est contenue dans ladite unité temporelle ; et
    ladite carte de points d’entrée contient soit un intervalle qui est défini depuis ladite adresse de tête de l’image
    I jusqu’à une adresse de fin de ladite image I, soit un intervalle qui est défini depuis ladite adresse de tête de
    l’image I jusqu’à une adresse de fin d’une image P qui fait suite à ladite adresse de fin de l’image I, comme
    contenu dans une unité temporelle incluant ladite image I.
12. Appareil selon la revendication 9, 10 ou 11, dans lequel le générateur de formation de carte de points d’entrée (16) fonctionne de telle sorte que ladite carte de points d’entrée inclue un indicateur pour indiquer si oui ou non ledit point d’entrée à l’intérieur de ladite unité temporelle est présent.

13. Appareil selon l’une quelconque des revendications 9 à 12 lorsqu’elle dépend de la revendication 7, dans lequel le générateur de formation de carte de points d’entrée (16) fonctionne pour générer une dite carte de points d’entrée respective pour chacun des programmes souhaités.

14. Appareil selon l’une quelconque des revendications 9 à 13, dans lequel le générateur de formation de carte de points d’entrée fonctionne pour modifier ladite carte de points d’entrée lorsque ledit train codé est édité.

15. Appareil selon l’une quelconque des revendications précédentes, comprenant un moyen d’archivage (17) pour archiver la carte d’unités temporelles en connexion avec les paquets de source générés.

16. Appareil selon l’une quelconque des revendications précédentes, comprenant un moyen d’enregistrement (20) qui peut fonctionner pour enregistrer lesdits paquets de source en association avec ladite carte d’unités temporelles sur un support d’enregistrement (21).

17. Procédé de traitement de données pour traiter des données qui sont contenues dans un train codé entré comprenant des paquets de transport, le procédé étant caractérisé par les étapes de :

- génération (14) de timbres temporels dont chacun est indicatif d’un temps d’arrivée de l’un associé des paquets de transport du train codé entré ;
- génération (15) de paquets de source en additionnant les timbres temporels d’arrivée aux paquets de transport associés ; et
- en réponse auxdits timbres temporels, génération (23) d’une carte d’unités temporelles qui comprend des adresses (A(TUi)) de chacune d’une pluralité d’unités temporelles successives (TUi) successives de durées identiques selon lesquelles le train codé entré est segmenté.

18. Procédé selon la revendication 17, dans lequel l’étape de génération de carte d’unités temporelles (23) inclut, dans la carte d’unités temporelles, une information qui est indicative de la quantité de données dans chaque unité temporelle (TU).

19. Procédé selon la revendication 18, dans lequel l’étape de génération de carte d’unités temporelles (23) génère une carte d’unités temporelles qui comprend l’adresse (A(TUi)) d’un paquet de tête de chaque dite unité temporelle (TU) et un intervalle (par exemple A(TU1)-A(TUO)) entre des adresses correspondant à la quantité de données dans chaque unité temporelle.

20. Procédé selon la revendication 18, dans lequel l’étape de génération de carte d’unités temporelles (23) inclut, dans la carte d’unités temporelles, une information de décalage qui est indicative de valeurs de décalage des paquets de source, et des adresses indiquant un numéro relatif du paquet de source par rapport à la valeur de décalage.

21. Procédé selon l’une quelconque des revendications 17 à 20, comprenant une étape (24) de production de numéros de paquet, l’étape de génération de carte d’unités temporelles (23) étant sensible auxdits numéros de paquet de même qu’auxdits timbres temporels pour générer la carte d’unités temporelles.

22. Procédé selon l’une quelconque des revendications 17 à 21, pour traiter des données qui sont contenues dans un train codé entré comprenant des paquets de transport qui concernent ceux respectifs d’une pluralité de programmes, dans lequel les paquets de transport qui concernent un programme souhaité sont sélectionnés (11) à partir du train codé et l’étape de génération de paquet de source (15) comprend la génération de paquets de source seulement pour les paquets de transport sélectionnés.

23. Procédé selon la revendication 22, dans lequel les paquets de transport qui concernent plusieurs programmes souhaités sont sélectionnés (11) à partir du train codé et une dite carte d’unités temporelles respective est générée (23) pour chacun des programmes souhaités.

24. Procédé selon l’une quelconque des revendications 17 à 23, dans lequel le contenu de ladite carte d’unités temporelles est modifié lorsque ledit train codé est édité.
25. Procédé selon l’une quelconque des revendications 17 à 24, comprenant la génération (16) d’une carte de points d’entrée qui est subordonnée à ladite carte d’unités temporelles, ladite carte de points d’entrée indiquant une position d’un point d’entrée dudit train codé.

26. Procédé selon la revendication 25, dans lequel ladite carte de points d’entrée inclut un intervalle d’adresses qui est défini à partir de données de tête de ladite unité temporelle du point d’entrée.

27. Procédé selon la revendication 26, dans lequel :

ledit point d’entrée représente une adresse de tête d’une image I qui est contenue dans ladite unité temporelle ; et ladite carte de points d’entrée contient soit un intervalle qui est défini depuis ladite adresse de tête de l’image I jusqu’à une adresse de fin de ladite image I, soit un intervalle qui est défini depuis ladite adresse de tête de l’image I jusqu’à une adresse de fin d’une image P qui fait suite à ladite adresse de fin de l’image I, comme contenu à l’intérieur d’une unité temporelle incluant ladite image I.

28. Procédé selon la revendication 25, 26 ou 27, dans lequel ladite carte de points d’entrée inclut un indicateur pour indiquer si oui ou non ledit point d’entrée à l’intérieur de ladite unité temporelle est présent.

29. Procédé selon l’une quelconque des revendications 25 à 28 lorsqu’elle dépend de la revendication 23, comprenant la génération d’une dite carte de points d’entrée respective pour chacun des programmes souhaités.

30. Procédé selon l’une quelconque des revendications 25 à 29, dans lequel ladite carte de points d’entrée est modifiée lorsque ledit train codé est édité.

31. Procédé selon l’une quelconque des revendications 17 à 30, comprenant l’archivage (17) de la carte d’unités temporelles en connexion avec les paquets de source générés.

32. Procédé selon l’une quelconque des revendications 17 à 31, comprenant l’enregistrement desdits paquets de source en association avec ladite carte d’unités temporelles sur un support d’enregistrement (21).

33. Support d’enregistrement comportant enregistré sur lui un programme lisible par ordinateur pour traiter des données qui sont contenues dans un train codé entré comprenant des paquets de transport, le programme comprenant les étapes selon l’une quelconque des revendications 17 à 32.

34. Appareil pour reproduire, à partir d’un support d’enregistrement, des paquets de source et une carte d’unités temporelles enregistrés sur le support au moyen d’un procédé selon la revendication 32, l’appareil comprenant un moyen (61-71) qui peut fonctionner pour reproduire la carte d’unités temporelles à partir du support d’enregistrement puis pour reproduire les paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite.

35. Appareil selon la revendication 34, pour reproduire, à partir d’un support d’enregistrement, des paquets de source, une carte d’unités temporelles et une carte de points d’entrée enregistrés sur le support au moyen d’un procédé selon la revendication 32 lorsqu’elle dépend de l’une quelconque des revendications 25 à 28, l’appareil comprenant un moyen (61-71) qui peut fonctionner pour reproduire la carte d’unités temporelles et la carte de points d’entrée à partir du support d’enregistrement puis pour reproduire les paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite et de la carte de points d’entrée reproduite.

36. Procédé de reproduction, à partir d’un support d’enregistrement, de paquets de source et d’une carte d’unités temporelles qui sont enregistrés sur le support au moyen d’un procédé selon la revendication 32, le procédé de reproduction comprenant les étapes de reproduction de la carte d’unités temporelles à partir du support d’enregistrement puis de reproduction des paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite et de la carte de points d’entrée reproduite.

37. Procédé selon la revendication 36 pour reproduire, à partir d’un support d’enregistrement, des paquets de source, une carte d’unités temporelles et une carte de points d’entrée qui sont enregistrés sur le support au moyen d’un procédé selon la revendication 32 lorsqu’elle dépend de l’une quelconque des revendications 25 à 28, le procédé de reproduction comprenant la reproduction de la carte d’unités temporelles et de la carte de points d’entrée à partir
du support d’enregistrement puis la reproduction des paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite et de la carte de points d’entrée reproduite.

38. Support d’enregistrement comportant enregistré sur lui un programme lisible par ordinateur pour reproduire, à partir d’un support d’enregistrement, des paquets de source et une carte d’unités temporelles enregistrés sur le support au moyen d’un procédé selon la revendication 32, le programme comprenant les étapes de reproduction de la carte d’unités temporelles à partir du support d’enregistrement puis de reproduction des paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite.

39. Support d’enregistrement comportant enregistré sur lui un programme lisible par ordinateur pour reproduire, à partir d’un support d’enregistrement, des paquets de source, une carte d’unités temporelles et une carte de points d’entrée enregistrés sur le support au moyen d’un procédé selon la revendication 32 lorsqu’elle dépend de l’une quelconque des revendications 25 à 28, le programme comprenant les étapes de reproduction de la carte d’unités temporelles et de la carte de points d’entrée à partir du support d’enregistrement puis de reproduction de paquets de source à partir d’une position arbitraire sur le support d’enregistrement sur la base de la carte d’unités temporelles reproduite et de la carte de points d’entrée reproduite.

40. Support d’enregistrement comportant enregistrés sur lui :
   des paquets de source qui sont générés en additionnant, à des paquets de transport d’un train de données codé, des timbres temporels d’arrivée dont chacun est indicatif d’un temps d’arrivée de l’un associé des paquets de transport du train codé ; et
   une carte d’unités temporelles qui comprend des adresses (A(TUi)) de chacune d’une pluralité d’unités temporelles successives (TUj) de durées identiques selon lesquelles le train codé est segmenté.

41. Support d’enregistrement selon la revendication 40, dans lequel la carte d’unités temporelles inclut une information qui est indicative de la quantité de données dans chaque unité temporelle (TUj).

42. Support d’enregistrement selon la revendication 41, dans lequel la carte d’unités temporelles comprend l’adresse (A(TUi)) d’un paquet de tête de chaque dite unité temporelle (TUj) et un intervalle (par exemple A(TU1)-A(TUO)) entre des adresses correspondant à la quantité de données dans chaque unité temporelle.

43. Support d’enregistrement selon la revendication 41, dans lequel la carte d’unités temporelles inclut une information de décalage qui est indicative de valeurs de décalage des paquets de source et des adresses indiquant un numéro relatif du paquet de source par rapport à la valeur de décalage.

44. Support d’enregistrement selon l’une quelconque des revendications 40 à 43, dans lequel la carte d’unités temporelles comprend une information qui est dérivée à partir de numéros de paquets de même que lesdits timbres temporels.

45. Support d’enregistrement selon l’une quelconque des revendications 40 à 44, dans lequel les paquets de source ont été générés seulement pour ceux sélectionnés des paquets de transport du train codé, lesquels concernent ceux respectifs d’une pluralité de programmes, les paquets de transport sélectionnés se rapportant à l’un souhaité de la pluralité de programmes.

46. Support d’enregistrement selon l’une quelconque des revendications 40 à 44, dans lequel les paquets de source ont été générés seulement pour ceux sélectionnés des paquets de transport du train codé, lesquels concernent ceux respectifs d’une pluralité de programmes, les paquets de transport sélectionnés se rapportant à l’un souhaité de plusieurs numéros de la pluralité de programmes.

47. Support d’enregistrement selon l’une quelconque des revendications 40 à 46, qui comporte enregistrée sur lui une carte de points d’entrée qui est subordonnée à ladite carte d’unités temporelles, ladite carte de points d’entrée indiquant une position d’un point d’entrée dudit train codé.

48. Support d’enregistrement selon la revendication 47, dans lequel ladite carte de points d’entrée inclut un intervalle d’adresses défini à partir de données de tête de ladite unité temporelle du point d’entrée.
EP 1 051 027 B1

49. Support d’enregistrement selon la revendication 48, dans lequel :

ledit point d’entrée représente une adresse de tête d’une image I qui est contenue dans ladite unité temporelle ; et
ladite carte de points d’entrée contient soit un intervalle qui est défini depuis ladite adresse de tête de l’image I jusqu’à une adresse de fin de ladite image I, soit un intervalle qui est défini depuis ladite adresse de tête de l’image I jusqu’à une adresse de fin d’une image P qui fait suite à ladite adresse de fin de l’image I comme contenu à l’intérieur d’une unité temporelle incluant ladite image I.

50. Support d’enregistrement selon la revendication 47, 48 ou 49, dans lequel ladite carte de points d’entrée inclut un indicateur pour indiquer si oui ou non ledit point d’entrée à l’intérieur de ladite unité temporelle est présent.

51. Support d’enregistrement selon l’une quelconque des revendications 47 à 50 lorsqu’elle dépend de la revendication 46, comprenant ladite carte de points d’entrée respective pour chacun des programmes souhaités.
FIG. 1A

INPUT TRANSPORT STREAM

SELECTED TRANSPORT PACKET

NON-SELECTED TRANSPORT PACKET

FIG. 1B

TRANSPORT PACKET OF SELECTED AV PROGRAM

TU0

Δt

TU1

2Δt

TU2

end_time = 3Δt

start_time = 0
**FIG. 3A**

INPUT TRANSPORT STREAM

**FIG. 3B**

TRANSPORT PACKET OF SELECTED AV PROGRAM

- **TU0**: start_time = 0
- **TU1**: Δt
- **TU2**: 2*Δt
- **TU3**: end_time = 3*Δt
FIG. 4

FIG. 5

<table>
<thead>
<tr>
<th>SOURCE PACKET</th>
<th>NUMBER OF BITS</th>
<th>MNEMONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_packet() {</td>
<td>}</td>
<td></td>
</tr>
<tr>
<td>TP_extra_header()</td>
<td>4 BYTES</td>
<td></td>
</tr>
<tr>
<td>transport_packet()</td>
<td>188 BYTES</td>
<td></td>
</tr>
</tbody>
</table>
FIG. 6

TP_extra_header

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>NUMBER OF BITS</th>
<th>MNEMONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP_extra_header () {</td>
<td></td>
<td></td>
</tr>
<tr>
<td>copy_permission_indicator</td>
<td>2</td>
<td>uim HBO</td>
</tr>
<tr>
<td>arrival_time_stamp</td>
<td>30</td>
<td>uim HBO</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIG. 7

TIME UNIT MAP

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>time_unit_address</th>
<th>delta_time_unit_address</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>A (TU0)</td>
<td>A (TU1) – A (TU0)</td>
</tr>
<tr>
<td>TU1</td>
<td>A (TU1)</td>
<td>A (TU2) – A (TU1)</td>
</tr>
<tr>
<td>TU2</td>
<td>A (TU2)</td>
<td>end_address – A (TU2)</td>
</tr>
</tbody>
</table>
FIG. 8

DATA OF TU0  DATA OF TU1  DATA OF TU2
A (TU0)  A (TU1)  A (TU2)

l_start_address  l_start_address

FIG. 9

ENTRY POINT MAP

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>entry_point_flag</th>
<th>l_start_offset_from_time_unit_address</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>TU1</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>TU2</td>
<td>1</td>
<td>b</td>
</tr>
</tbody>
</table>
FIG. 10

ADDRESS

time_unit_address

I_start_offset_from_time_unit_address

I_end_offset_address

I_end_address

P1_end_offset_address

P1_end_address

P2_end_offset_address

P2_end_address
**FIG. 11A**

DATA OF TU0  DATA OF TU1  DATA OF TU2

---

A (TU0)  A (TU1)  A (TU2)

DELETE  DELETE

C  D

ADDRESS

---

**FIG. 11B**

DELETED DATA OF TU0

DATA OF TU1  DELETED DATA OF TU2

---

C  A (TU1)  A (TU2)  D

ADDRESS

---

**FIG. 12**

TIME UNIT MAP AFTER PARTIAL DELETION

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>delta_time_unit_address</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>A (TU1) – C</td>
</tr>
<tr>
<td>TU1</td>
<td>A (TU2) – A (TU1)</td>
</tr>
<tr>
<td>TU2</td>
<td>D – A (TU2)</td>
</tr>
</tbody>
</table>
FIG. 13

TimeUnitMapHeader ()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  TimeUnitMapHeader () {</td>
<td></td>
</tr>
<tr>
<td>2  start_time</td>
<td>N1</td>
</tr>
<tr>
<td>3  end_time</td>
<td>N2</td>
</tr>
<tr>
<td>4  first_time_unit_size</td>
<td>N3</td>
</tr>
<tr>
<td>5  time_unit_size</td>
<td>N4</td>
</tr>
<tr>
<td>6  number_of_time_unit_entries</td>
<td>N5</td>
</tr>
<tr>
<td>7  }</td>
<td></td>
</tr>
</tbody>
</table>

FIG. 14

TimeUnitMapData ()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  TimeUnitMapData () {</td>
<td></td>
</tr>
<tr>
<td>2  for (i = 0 ; i &lt; number_of_time_unit_entries ; i ++ ) {</td>
<td></td>
</tr>
<tr>
<td>3  delta_time_unit_address</td>
<td>N6</td>
</tr>
<tr>
<td>4  }</td>
<td></td>
</tr>
<tr>
<td>5  }</td>
<td></td>
</tr>
</tbody>
</table>
**FIG. 15**

EntryPointMapHeader()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EntryPointMapHeader () {</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>number_of_programs</td>
<td>K1</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>for (i = 0 ; i &lt; number_of_programs ; i ++ ) {</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>program_number</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>parsed_program_flag</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>for (i = 0 ; i &lt; NUMBER_OF_ParsedPrograms ; i ++ ) {</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>MPEG_TS_program_map_section ()</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 16**

EntryPointMapData()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EntryPointMapData () {</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>for (i = 0 ; i &lt; NUMBER_OF_ParsedPrograms ; i ++ ) {</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>for (j = 0 ; j &lt; number_of_time_unit_entries ; j ++ )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>entry_point_flag</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>entry_point_data ()</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
FIG. 17

entry_point_data ()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 entry_point_data () {</td>
<td></td>
</tr>
<tr>
<td>2 entry_point_time_stamp</td>
<td>K2</td>
</tr>
<tr>
<td>3 l_start_offset_from_time_unit_address</td>
<td>K3</td>
</tr>
<tr>
<td>4 l_end_offset_address</td>
<td>K4</td>
</tr>
<tr>
<td>5 P1_end_offset_address</td>
<td>K5</td>
</tr>
<tr>
<td>6 P2_end_offset_address</td>
<td>K6</td>
</tr>
<tr>
<td>7 }</td>
<td></td>
</tr>
</tbody>
</table>

FIG. 18

EntryPointMapData ()

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 EntryPointMapData () {</td>
<td></td>
</tr>
<tr>
<td>2 for (i = 0 ; i &lt; number_of_time_unit_entries ; i + + ) {</td>
<td></td>
</tr>
<tr>
<td>3 for (j = 0 ; j &lt; NUMBER_OF_ParsedPrograms ; j + + )</td>
<td></td>
</tr>
<tr>
<td>4 entry_point_flag</td>
<td>1</td>
</tr>
<tr>
<td>5 entry_point_data ()</td>
<td></td>
</tr>
<tr>
<td>6 }</td>
<td></td>
</tr>
<tr>
<td>7 }</td>
<td></td>
</tr>
<tr>
<td>8 }</td>
<td></td>
</tr>
</tbody>
</table>
FIG. 19

DATA OF TU0  DATA OF TU1  DATA OF TU2  DATA OF TU3

ENTRY POINT 1  ENTRY POINT 2  ENTRY POINT 3  ENTRY POINT 4
OF PROGRAM #1 OF PROGRAM #1 OF PROGRAM #1 OF PROGRAM #1

ENTRY POINT 1  ENTRY POINT 2
OF PROGRAM #2 OF PROGRAM #2

ENTRY POINT 1  ENTRY POINT 2
OF PROGRAM #3 OF PROGRAM #3

ADDRESS
### FIG. 20A

**EntryPointMapData** of program #1

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>entry_point_flag</th>
<th>entry_point_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>1</td>
<td>entry_point_data#1-1</td>
</tr>
<tr>
<td>TU1</td>
<td>1</td>
<td>entry_point_data#1-2</td>
</tr>
<tr>
<td>TU2</td>
<td>1</td>
<td>entry_point_data#1-3</td>
</tr>
<tr>
<td>TU3</td>
<td>1</td>
<td>entry_point_data#1-4</td>
</tr>
</tbody>
</table>

### FIG. 20B

**EntryPointMapData** of program #2

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>entry_point_flag</th>
<th>entry_point_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>1</td>
<td>entry_point_data#2-1</td>
</tr>
<tr>
<td>TU1</td>
<td>0</td>
<td>– (dummy data)</td>
</tr>
<tr>
<td>TU2</td>
<td>1</td>
<td>entry_point_data#2-2</td>
</tr>
<tr>
<td>TU3</td>
<td>0</td>
<td>–</td>
</tr>
</tbody>
</table>

### FIG. 20C

**EntryPointMapData** of program #3

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>entry_point_flag</th>
<th>entry_point_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>TU1</td>
<td>1</td>
<td>entry_point_data#3-1</td>
</tr>
<tr>
<td>TU2</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>TU3</td>
<td>1</td>
<td>entry_point_data#3-2</td>
</tr>
</tbody>
</table>
**FIG. 21**

EntryPointMapData OF program #1, program #2 AND program #3

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>PROGRAM NUMBER</th>
<th>entry_point_flag</th>
<th>entry_point_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>#1</td>
<td>1</td>
<td>entry_point_data#1-1</td>
</tr>
<tr>
<td></td>
<td>#2</td>
<td>1</td>
<td>entry_point_data#2-1</td>
</tr>
<tr>
<td></td>
<td>#3</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>TU1</td>
<td>#1</td>
<td>1</td>
<td>entry_point_data#1-2</td>
</tr>
<tr>
<td></td>
<td>#2</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>#3</td>
<td>1</td>
<td>entry_point_data#3-1</td>
</tr>
<tr>
<td>TU2</td>
<td>#1</td>
<td>1</td>
<td>entry_point_data#1-3</td>
</tr>
<tr>
<td></td>
<td>#2</td>
<td>1</td>
<td>entry_point_data#2-2</td>
</tr>
<tr>
<td></td>
<td>#3</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>TU3</td>
<td>#1</td>
<td>1</td>
<td>entry_point_data#1-4</td>
</tr>
<tr>
<td></td>
<td>#2</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>#3</td>
<td>1</td>
<td>entry_point_data#3-2</td>
</tr>
</tbody>
</table>
FIG. 22A
INPUT TRANSPORT STREAM

FIG. 22B
TRANSPORT PACKET OF SELECTED AV PROGRAM

FIG. 22C
RECORDED SOURCE PACKET (DATA OF TRANSPORT STREAM ON RECORDING MEDIUM)

FIG. 22D
TIME UNIT MAP

<table>
<thead>
<tr>
<th>TIME UNIT</th>
<th>RSPN_time_unit_start</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>A (TU0) = 0</td>
<td>ZEROTH TIME UNIT</td>
</tr>
<tr>
<td>TU1</td>
<td>A (TU1) = 4</td>
<td>FIRST TIME UNIT</td>
</tr>
<tr>
<td>TU2</td>
<td>A (TU2) = 7</td>
<td>SECOND TIME UNIT</td>
</tr>
</tbody>
</table>
**FIG. 23**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>NUMBER OF BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeUnitMapHeader () {</td>
<td></td>
</tr>
<tr>
<td>offset_SPN</td>
<td>32</td>
</tr>
<tr>
<td>offset_time</td>
<td>32</td>
</tr>
<tr>
<td>time_unit_size</td>
<td>32</td>
</tr>
<tr>
<td>number_of_time_unit_entries</td>
<td>32</td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 24**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>NUMBER OF BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeUnitMapData () {</td>
<td></td>
</tr>
<tr>
<td>for (k = 0 ; k &lt; number_of_time_unit_entries ; k ++ ) {</td>
<td></td>
</tr>
<tr>
<td>RSPN_time_unit_start</td>
<td>32</td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

{ }
**FIG. 26A**

**TIME UNIT MAP BEFORE BEING PARTIALLY DELETED**

<table>
<thead>
<tr>
<th></th>
<th>RSPN_time_unit_start</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU0</td>
<td>A (TU0) = 0</td>
<td>ZEROTH TIME UNIT</td>
</tr>
<tr>
<td>TU1</td>
<td>A (TU1) = 4</td>
<td>FIRST TIME UNIT</td>
</tr>
<tr>
<td>TU2</td>
<td>A (TU2) = 7</td>
<td>SECOND TIME UNIT</td>
</tr>
</tbody>
</table>

- offset_SPN = 0
- offset_time = 0 ; (STARTING TIME INSTANT OF TU0)
- time_unit_size = ΔT
- number_of_time_unit_entries = 3

**FIG. 26B**

**TIME UNIT MAP AFTER BEING PARTIALLY DELETED**

<table>
<thead>
<tr>
<th></th>
<th>RSPN_time_unit_start</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU1</td>
<td>A (TU1) = 4</td>
<td>NEW ZEROTH TIME UNIT</td>
</tr>
<tr>
<td>TU2</td>
<td>A (TU2) = 7</td>
<td>NEW FIRST TIME UNIT</td>
</tr>
</tbody>
</table>

- offset_SPN = C (= 2)
- offset_time = ΔT ; (STARTING TIME INSTANT OF TU1)
- time_unit_size = ΔT
- number_of_time_unit_entries = 2
FIG. 28

START PAT/PMT ANALYZING PROCESS

RECEIVE PAT TO ACQUIRE PID OF PMT OF EACH PROGRAM ~ S11

SET PID OF PMT OF EACH PROGRAM TO PID FILTER ~ S12

RECEIVE PMT TO ACQUIRE PID OF VIDEO PACKET/AUDIO PACKET, AND PID OF PCR FOR CONSTITUTING EACH PROGRAM ~ S13

SET PID OF VIDEO PACKET/AUDIO PACKET, AND PID OF PCR FOR CONSTITUTING EACH PROGRAM TO PID FILTER ~ S14

END
FIG. 29

START ANALYZING PROCESS OF VIDEO STREAM

S31 ~ SET PID, stream_type OF VIDEO

S32 ~ vpp = (-1): INITIALIZE POINTER OF VIDEO PACKET

S33 ~ vpp++: INCREMENT POINTER OF VIDEO PACKET

S34 ~ sequence_header_code IS CONTAINED?

YES

S35 ~ SET PACKET CONTAINING sequence_header_code AS ADDRESS OF I_start

S36 ~ vpp++

S37 ~ I PICTURE DATA IS ENDED?

NO

YES

S38 ~ SET PACKET FOR ENDING I PICTURE DATA AS ADDRESS OF I_end
FIG. 30

1

NEXT VIDEO PACKET CONTAINS sequence_header_code ?

3

YES

S39

NO

S40 \( \sim \) vpp++

S41 EITHER P PICTURE OR I PICTURE IS ENDED?

YES

NO

S42

SET PACKET FOR ENDING EITHER P PICTURE OR I PICTURE AS P1_end ADDRESS

3

YES

S43

NO

S44 \( \sim \) vpp++

S45 EITHER P PICTURE OR I PICTURE IS ENDED?

3

YES

NO

S46

SET PACKET FOR ENDING EITHER P PICTURE OR I PICTURE AS P2_end ADDRESS

3

INPUT ADDRESSES OF l_start, l_end, P1_end AND P2_end INTO ENTRY POINT MAP FORMING UNIT

3

LAST PACKET?

NO

S47

YES

S48

END

2

51
FIG. 34

START HIGH SPEED PROCESSING OPERATION

S61 SET program_number

S62 parsed_program_flag == 1?

NO

YES

CALCULATE NUMBER TN OF TIME UNIT FOR STARTING READING OPERATION FROM DESIGNATED REPRODUCTION STARTING TIME INSTANT

S63

S64 entry_point_flag == 1?

NO

YES

CALCULATE ADDRESS OF ENTRY POINT

S65

INSTRUCT READING UNIT TO READ STREAM DATA OF ENTRY POINT

S66

INCREMENT TN

S67

PROCESSING OPERATION IS ENDED?

NO

S68

YES

END
FIG. 35

START PROCESSING OPERATION WHEN PROGRAM NUMBER AND REPRODUCING TIME INSTANT ARE DESIGNATED TO REPRODUCE PROGRAM

INPUT program_number AND REPRODUCING START TIME INSTANT Tst FROM USER INTERFACE TO REPRODUCING CONTROL UNIT

S81

Tst = start_time?

S82

YES

NO

S83

REPRODUCING CONTROL UNIT READS HEADER PORTION OF TIME UNIT MAP

S86

N = 0,

time_unit_address (N) = 0

S84

REPRODUCING CONTROL UNIT CALCULATES MINIMUM N CAPABLE OF SATISFYING BELOW-MENTIONED INEQUALITY

Tst ≤ start_time + first_time_unit_size + (N - 1) x time_unit_size

S85

REPRODUCING CONTROL UNIT CALCULATES time_unit_address (N) WHICH IS CALCULATED BASED ON DATA OF TIME UNIT MAP BY BELOW-MENTIONED EQUATION

\[ time_unit_address (N) = \sum_{i=0}^{N-1} \delta \text{time_unit_address}(i) \]

S87

REPRODUCING CONTROL UNIT INSTRUCTS READING UNIT TO READ DATA FROM ADDRESS time_unit_address (N) OF N-TH TIME UNIT
FIG. 36

1

READING UNIT READ TRANSPORT STREAM FROM ADDRESS "time_unit_address (N)"

S88

REPRODUCING CONTROL UNIT INPUTS program_number INTO DEMULTIPLEXER

S89

SEPARATE TRANSPORT PACKET OF PROGRAM OF program_number DESIGNATED BY DEMULTIPLEXER

S90

AV DECODER DECODES PROGRAM

S91

END
FIG. 37

START PROCESSING OPERATION WHEN PROGRAM NUMBER AND REPRODUCING TIME INSTANT ARE DESIGNATED TO REPRODUCE PROGRAM

INPUT program_number AND REPRODUCING START TIME INSTANT Tst FROM USER INTERFACE TO REPRODUCING CONTROL UNIT

S81

S102

Tst ！= offset_time? NO

YES

S83

REPRODUCING CONTROL UNIT READS HEADER PORTION OF TIME UNIT MAP

S86

N = 0

time_unit_address (N) = 0

REPRODUCING CONTROL UNIT CALCULATES MINIMUM N CAPABLE OF SATISFYING BELOW-MENTIONED INEQUALITY

Tst ≤ offset_time + N*time_unit_size

S104

REPRODUCING CONTROL UNIT CALCULATED time_unit_address (N) WHICH IS CALCULATED BASED ON DATA OF TIME UNIT MAP BY BELOW-MENTIONED EQUATION

time_unit_address (N) = (RSPN_time_unit_start (N) - offset_SPN)

S105

REPRODUCING CONTROL UNIT INSTRUCTS READING UNIT TO READ DATA FROM ADDRESS "time_unit_address (N)" OF N-TH TIME UNIT

S87

S88

READING UNIT READS TRANSPORT STREAM FROM ADDRESS "time_unit_address (N)"

S89

REPRODUCING CONTROL UNIT INPUTS program_number INTO DEMULTIPLEXER

S90

SEPARATE TRANSPORT PACKET OF PROGRAM OF program_number DESIGNATED BY DEMULTIPLEXER

S91

AV DECODER DECODES PROGRAM

END